Piecewise Parametric Structure in the Pooling Problem – from Sparse Strongly-Polynomial Solutions to NP-Hardness

Radu Baltean-Lugojan
radu.baltean-lugojan09@imperial.ac.uk

Ruth Misener
r.misener@imperial.ac.uk

Computational Optimisation Group
Centre for Process Systems Engineering
Department of Computing

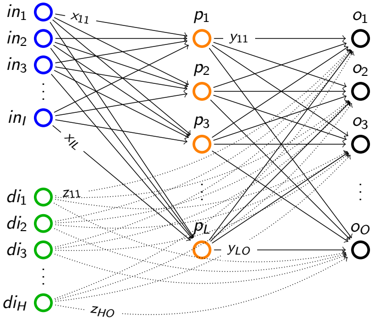
Imperial College London

computational optimisation group

Centre for Process Systems Engineering

2017/05/18

Globally Optimizing Pooling Problems



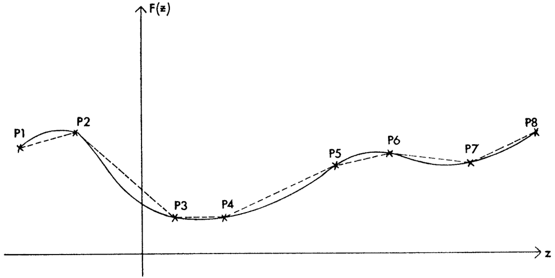
Previous work

- **Explore P / NP boundary** Alfaki and Haugland [2013] • Haugland [2016] • Haugland and Hendrix [2016] • Boland et al. [2016]
- **MIP Approximation** Dey and Gupte [2015] • Gupte et al. [2016]
- **Cutting Planes** D'Ambrosio et al. [2011] • Gupte et al. [2016]
- **Finding Patterns** Ceccon et al. [2016]

This work: Parameterize with respect to pool concentration

- Develop **strongly** polynomial algorithms for several pooling subproblems;
- Use **patterns of dominating topologies** to find active network structure and motivate problem sparsity.

How to solve problems multiplying quality times flowrate?



Haverly [1978]

Solve pooling problem via sequential linear programming, i.e., approximate nonconvex bilinear terms as linear.

Beale et al. [1965]

- **In Iron-Making** Track manganese concentration between two blast furnaces and a sinter plant;
- **To Solve** Reformulate as a separable program and apply a piecewise linear approximation;
- **Key Assumption** Only a few variables are active at once and the associated functions can be approximated linearly.

Standard Pooling Network p-Formulation

Objective $\max_{x_{il}, y_{lj}, z_{ij}, p_{lk}} \sum_{(l,j)\in T_Y} d_j \cdot y_{lj} + \sum_{(i,j)\in T_Z} d_j \cdot z_{ij} - \sum_{(i,l)\in T_X} \gamma_i \cdot x_{il} - \sum_{(i,j)\in T_Z} \gamma_i \cdot z_{ij}$

Feed Avail $\left[A_i^L \leq \sum_{l:(i,l)\in T_X} x_{il} + \sum_{j:(i,j)\in T_Z} z_{ij} \leq A_i^U \quad \forall\, i \right.$

Pool Capacity $\left[S_l^L \leq \sum_{i:(i,l)\in T_X} x_{il} \leq S_l^U \quad \forall\, l \right.$

Product Demand $\left[D_j^L \leq \sum_{l:(l,j)\in T_Y} y_{lj} + \sum_{i:(i,j)\in T_Z} z_{ij} \leq D_j^U \quad \forall\, j \right.$

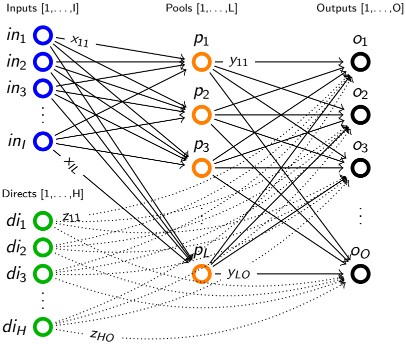
Material Balance $\left[\sum_{i:(i,l)\in T_X} x_{il} - \sum_{j:(l,j)\in T_Y} y_{lj} = 0 \quad \forall\, l \right.$

Quality Balance $\left[\sum_{i:(i,l)\in T_X} C_{ik} x_{il} = p_{lk} \sum_{j:(l,j)\in T_Y} y_{lj} \quad \forall\, l, k \right.$

Product Quality

$$\left[\begin{array}{l} \displaystyle\sum_{l:(l,j)\in T_Y} p_{lk} y_{lj} \\ \displaystyle + \sum_{i:(i,j)\in T_Z} C_{ik} z_{ij} \end{array} \right. \begin{cases} \geq P_{jk}^L \left(\displaystyle\sum_{l:(l,j)\in T_Y} y_{lj} + \sum_{i:(i,j)\in T_Z} z_{ij} \right) \\ \leq P_{jk}^U \left(\displaystyle\sum_{l:(l,j)\in T_Y} y_{lj} + \sum_{i:(i,j)\in T_Z} z_{ij} \right) \end{cases} \quad \forall\, j, k$$

Bounds $[x_{il},\ y_{lj},\ z_{ij} \geq 0 \ \forall\ i,\ l,\ j$



Standard Pooling Network Restricted p-Formulation

Objective $\max_{x_{il}, y_{lj}, z_{ij}, p_l} \sum_{(l,j)\in T_Y} d_j \cdot y_{lj} + \sum_{(i,j)\in T_Z} d_j \cdot z_{ij} - \sum_{(i,l)\in T_X} \gamma_i \cdot x_{il} - \sum_{(i,j)\in T_Z} \gamma_i \cdot z_{ij}$

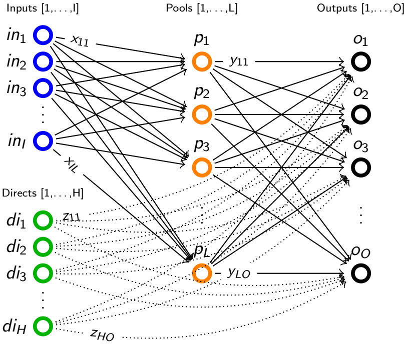
Product Demand $\left[\sum_{l:(l,j)\in T_Y} y_{lj} + \sum_{i:(i,j)\in T_Z} z_{ij} \leq D_j^U \quad \forall\, j \right.$

Material Balance $\left[\sum_{i:(i,l)\in T_X} x_{il} - \sum_{j:(l,j)\in T_Y} y_{lj} = 0 \quad \forall\, l \right.$

Quality Balance $\left[\sum_{i:(i,l)\in T_X} C_i x_{il} = p_l \sum_{j:(l,j)\in T_Y} y_{lj} \quad \forall\, l \right.$

Product Quality
$$\left[\begin{array}{l} \displaystyle\sum_{l:(l,j)\in T_Y} p_l y_{lj} \\ \displaystyle + \sum_{i:(i,j)\in T_Z} C_i z_{ij} \end{array} \right. \begin{cases} \geq P_j^L \left(\displaystyle\sum_{l:(l,j)\in T_Y} y_{lj} + \sum_{i:(i,j)\in T_Z} z_{ij} \right) \\ \leq P_j^U \left(\displaystyle\sum_{l:(l,j)\in T_Y} y_{lj} + \sum_{i:(i,j)\in T_Z} z_{ij} \right) \end{cases} \quad \forall\, j$$

Bounds $\left[x_{il},\ y_{lj},\ z_{ij} \geq 0 \ \forall\ i,\ l,\ j \right.$



Assumptions

- No bounds on feed availability or pool capacity. No lower bounds and finite upper bounds on product demand. Only 1 quality k.
- The problem is feasible and profitable ($f^* > 0$)
- **Known:** NP-hard [Haugland, 2016]

Restricted p-Formulation for One Pool, One Output

Objective $\max_{\{x_i\},y,\{z_i\},p} f = d \cdot y + \sum_{i:(i,1)\in T_Z} d \cdot z_i - \sum_{i:(i,1)\in T_X} \gamma_i \cdot x_i - \sum_{i:(i,1)\in T_Z} \gamma_i \cdot z_i$

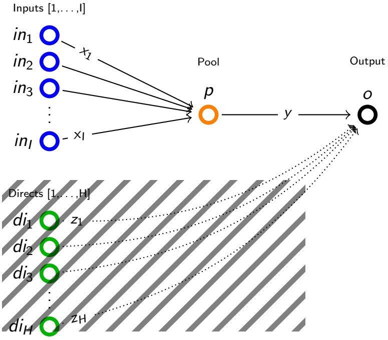
Product Demand $\left[y + \sum_{i:(i,1)\in T_Z} z_i \leq D^U \right.$

Material Balance $\left[\sum_{i:(i,1)\in T_X} x_i - y = 0 \right.$

Quality Balance $\left[\sum_{i:(i,1)\in T_X} C_i x_i = p \cdot y \right.$

Product Quality $\left[p \cdot y + \sum_{i:(i,1)\in T_Z} C_i \cdot z_i \right. \begin{cases} \geq \mathbf{P^L}\left(\mathbf{y} + \sum_{\mathbf{i:(i,1)\in T_Z}} \mathbf{z_i}\right) \\ \leq P^U\left(y + \sum_{i:(i,1)\in T_Z} z_i\right) \end{cases}$

Bounds $[x_i,\ y,\ z_i \geq 0\ \forall\ i$



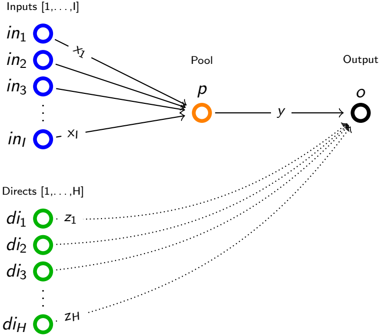
Known result

Can solve with a polynomial number of LPs of polynomial size [Haugland and Hendrix, 2016, Boland et al., 2016]

Restricted p-Formulation for One Pool, One Output

$$\max_{\{x_i\},\{z_i\}} f = d \cdot D^U - \sum_{i \in T_X} \gamma_i \cdot x_i - \sum_{i \in T_Z} \gamma_i \cdot z_i$$

$$\begin{cases} \sum_{i \in T_X} x_i + \sum_{i \in T_Z} z_i = D^U \\ P^L \le \left(\sum_{i \in T_X} C_i x_i + \sum_{i \in T_Z} C_i z_i \right) \Big/ \left(\sum_{i \in T_X} x_i + \sum_{i \in T_Z} z_i \right) \le P^U \\ p = \sum_{i \in T_X} C_i x_i \Big/ \sum_{i \in T_X} x_i \\ 0 \le x_i, \ \forall\ i \in T_X \qquad 0 \le z_i, \ \forall\ i \in T_Z \end{cases}$$



Solve by treating in turn:

- No directs (only inputs and quality constraints dropped)
- No inputs (only directs and quality constraints kept)
- Inputs + directs (quality constraints present)

Definition - Active set

Active set

Set A of nodes from the feed layer is:

- An **input active set** if $A \subseteq T_X,\ x_i > 0\ \forall i \in A,\ x_i = 0\ \forall i \in T_X \setminus A,\ z_i = 0\ \forall i \in T_Z.$
- A **direct active set** if $A \subseteq T_Z,\ z_i > 0\ \forall i \in A,\ z_i = 0\ \forall i \in T_Z \setminus A,\ x_i = 0\ \forall i \in T_X.$
- A **mixed active set** if $A \subseteq T_X \cup T_Z,\ x_i > 0\ \forall i \in A \cap T_X \neq \emptyset,\ z_i > 0\ \forall i \in A \cap T_Z \neq \emptyset, z_i = 0\ \forall i \in T_Z \setminus A,\ x_i = 0\ \forall i \in T_X \setminus A.$

For an active set A in One Pool-One Output, the objective function f,

$$f = \begin{cases} f_A(p) = dD - \sum_{i \in A \cap T_X} \gamma_i x_i - \sum_{i \in A \cap T_Z} \gamma_i z_i, & A \text{ an input/mixed active set,} \\ f_A = dD - \sum_{i \in A} \gamma_i z_i, & A \text{ a direct active set,} \end{cases}$$

Let $h = dD - f$ denote the cost function associated with objective function f.

Definition - Feasibility

Feasibility with respect to product quality constraints

A One Pool-One Output active set is **feasible** if the product quality bounds $[P^L, P^U]$ are met, i.e. the second constraint holds. An infeasible active set is not a valid One Pool-One Output solution and is therefore strictly dominated by any feasible active set.

Definition - Dominance

Dominance and breakpoints between active sets

Let A_1, A_2 be feasible input/mixed active sets. Let $f_A^*(p)$ be the optimal solution to One Pool-One Output and $h_A^*(p)$ its corresponding optimal cost, assuming active set A and fixed p.

- Set A_1 **dominates** A_2 **at** p (in the sense of maximized objective function profitability) when,

$$\boldsymbol{A}_1 \succeq_p \boldsymbol{A}_2 \quad \Leftrightarrow \quad f_{A_1}^*(p) \geq f_{A_2}^*(p) \quad \Leftrightarrow \quad h_{A_1}^*(p) \leq h_{A_2}^*(p).$$

- Pool concentration p is a **breakpoint between** A_1 **and** A_2 if:

$$\boldsymbol{A}_1 \asymp_p \boldsymbol{A}_2 \quad \Leftrightarrow \quad f_{A_1}^*(p) = f_{A_2}^*(p) \quad \Leftrightarrow \quad h_{A_1}^*(p) = h_{A_2}^*(p).$$

- The dominance relation also extends to direct active sets, but in this case f is not parametric on p. Consequently, when comparing two direct active sets, dominance is established similarly but independent of p, and as such no breakpoints exist. Thus, for fixed p, a total order can be established over the set of all possible active sets.

Definition - Dominant active sets

Dominant active sets and dominance breakpoints

Let $\mathcal{A}^*(p)$ be the **dominant active set** (overall) at p if

$$\mathcal{A}^*(p) = \underset{\{\mathcal{A}_I, \mathcal{A}_D, \mathcal{A}_M\}}{\arg\max} (f^*_{\mathcal{A}_I}(p), f^*_{\mathcal{A}_D}, f^*_{\mathcal{A}_M}(p)) = \underset{\{\mathcal{A}_I, \mathcal{A}_D, \mathcal{A}_M\}}{\arg\min} (h^*_{\mathcal{A}_I}(p), h^*_{\mathcal{A}_D}, h^*_{\mathcal{A}_M}(p))$$

and the optimal objective solution is $f^* = \max_p f^*_{\mathcal{A}^*(p)}(p)$, where:

- $\mathcal{A}_I$ is the **dominant input active set at** p if $\mathcal{A}_I(p) = \underset{A \subseteq T_X}{\arg\max} f^*_A(p)$.
- $\mathcal{A}_M$ is the **dominant mixed active set at** p if $\mathcal{A}_M(p) = \underset{A \subseteq T_X \cup T_Z,\ A \cap T_X \neq \emptyset,\ A \cap T_Z \neq \emptyset}{\arg\max} f^*_A(p)$.
- $\mathcal{A}_D$ is the **dominant direct active set** if $\mathcal{A}_D = \underset{A \subseteq T_Z}{\arg\max} f^*_A$.

A **dominance breakpoint** is the p value where the dominant active set changes:

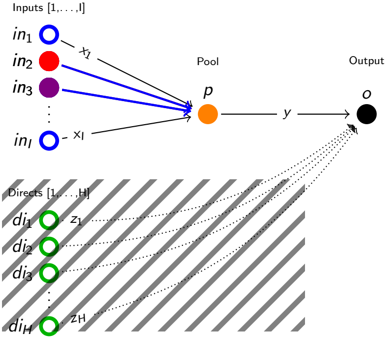
$$\forall \epsilon > 0 \ \mathcal{A}^*(p - \epsilon) = A_1 \neq A_2 = \mathcal{A}^*(p + \epsilon) \text{ and } A_1 \asymp_p A_2.$$

Input & mixed dominance breakpoints are similarly defined but $\mathcal{A}^*$ replaced by $\mathcal{A}_I$ & $\mathcal{A}_M$. Denote the input & mixed dominance breakpoints sets $\mathcal{B}_I$ & $\mathcal{B}_M$.

Only Input $\rightarrow$ Pool Flows (No Directs)

$$\max_{\{x_i\}} f = d \cdot D^U - \sum_{i \in T_X} \gamma_i \cdot x_i$$

$$\begin{cases} \sum_{i \in T_X} x_i = D^U \\ p = \sum_{i \in T_X} C_i x_i \Big/ \sum_{i \in T_X} x_i \\ 0 \leq x_i, \ \forall\, i \in T_X \end{cases}$$



For fixed p, consider 2 input nodes $\mathbf{in_2}$, $\mathbf{in_3}$ where $\mathbf{C_2} \leq p \leq \mathbf{C_3}$. Then the flows are:

$$\mathbf{x_2} = \frac{\mathbf{D^U(p - C_3)}}{\mathbf{C_2 - C_3}}, \quad \mathbf{x_3} = \frac{\mathbf{D^U(p - C_2)}}{\mathbf{C_3 - C_2}},$$

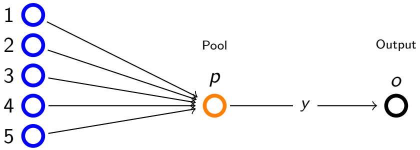
More generally, input pair $\{\mathbf{i}, \mathbf{j}\}$ has the best return on investment for a fixed p when:

$$\{\mathbf{i}, \mathbf{j}\} = \operatorname*{arg\,min}_{\mathbf{x} = \{\mathbf{i},\mathbf{j}\} \subseteq \mathbf{T_X}} \gamma_{\mathbf{x}}(\mathbf{p}) = \frac{\gamma_{\mathbf{i}}(\mathbf{p} - \mathbf{C_j}) + \gamma_{\mathbf{j}}(\mathbf{C_i} - \mathbf{p})}{\mathbf{C_i} - \mathbf{C_j}}$$

and $\{\mathbf{i}, \mathbf{j}\}$ is the **dominant input active set** (over any cardinality). The **breakpoint** in dominance of $\{\mathbf{i}, \mathbf{j}\}$ w.r.t. a given $\{\mathbf{k}, \mathbf{l}\}$ occurs at points

$$p_{\{\mathbf{i},\mathbf{j}\},\{\mathbf{k},\mathbf{l}\}} = \left(\frac{\gamma_{\mathbf{j}} \mathbf{C_i} - \gamma_{\mathbf{i}} \mathbf{C_j}}{\mathbf{C_i} - \mathbf{C_j}} - \frac{\gamma_{\mathbf{l}} \mathbf{C_k} - \gamma_{\mathbf{k}} \mathbf{C_l}}{\mathbf{C_k} - \mathbf{C_l}} \right) \Big/ \left(\frac{\gamma_{\mathbf{i}} - \gamma_{\mathbf{j}}}{\mathbf{C_i} - \mathbf{C_j}} - \frac{\gamma_{\mathbf{k}} - \gamma_{\mathbf{l}}}{\mathbf{C_k} - \mathbf{C_l}} \right) \forall \begin{cases} \mathbf{i}, \mathbf{j}, \mathbf{k}, \mathbf{l} \in T_X, \\ \{\mathbf{i}, \mathbf{j}\} \neq \{\mathbf{l}, \mathbf{k}\} \end{cases}$$

Only Input → Pool Flows: Numerical Example



$(C_1 = 1, \gamma_1 = 135)$, $(C_2 = 2, \gamma_2 = 105)$, $(C_3 = 3, \gamma_3 = 90)$, $(C_4 = 4, \gamma_4 = 111)$, $(C_5 = 5, \gamma_5 = 140)$; $D^U = 100$; $d = 150$;

$$\partial \mathbf{f} / \partial \mathbf{p} = -\mathbf{D}^{\mathbf{U}}(\gamma_{\mathbf{i}} - \gamma_{\mathbf{j}})/(\mathbf{C}_{\mathbf{i}} - \mathbf{C}_{\mathbf{j}})$$

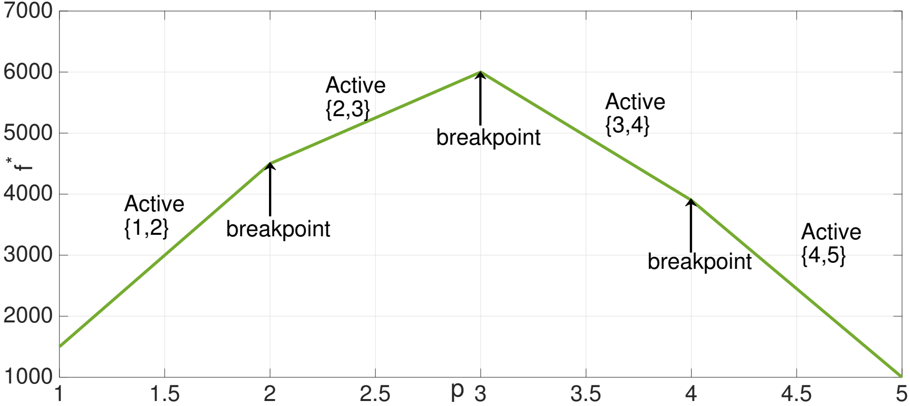
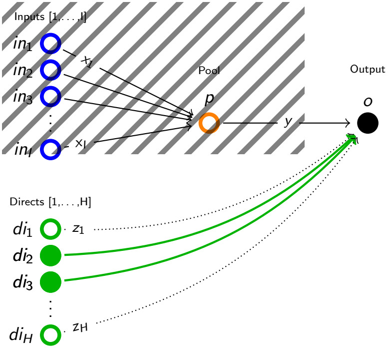


Figure 1: Optimal Objective Function $f^*(p)$ vs. pool concentration p

Only Input → Output Flows (No Pool)

$$\max_{\{z_i\}} f = d \cdot D^U - \sum_{i \in T_Z} \gamma_i \cdot z_i$$

$$\begin{cases} \sum_{i \in T_Z} z_i = D^U \\ P^L \le \sum_{i \in T_Z} C_i z_i \Big/ \sum_{i \in T_Z} z_i \le P^U \\ 0 \le z_i, \ \forall \ i \in T_Z \end{cases}$$



Given θ, the lowest cost direct node, i.e.,

$$\gamma_\theta = \min_{\mathbf{i} \in \mathbf{T_Z}} \gamma_{\mathbf{i}},$$

if $\mathbf{C}_\theta \in [\mathbf{P^L}, \mathbf{P^U}]$ then θ is the dominant direct active set.

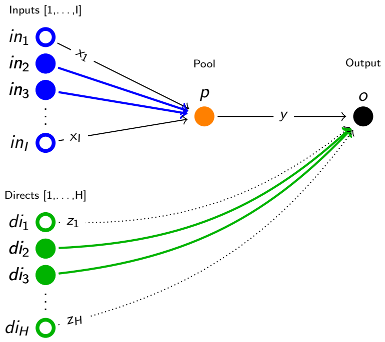
Otherwise a feasible active direct pair $\{\mathbf{i}, \mathbf{j}\}$ dominates any alternative:

$$\{\mathbf{i}, \mathbf{j}\} = \underset{\{\alpha,\beta\} \subseteq \mathbf{T_Z}}{\arg\min} \frac{\gamma_\alpha(\mathbf{P}^\star - \mathbf{C}_\beta) + \gamma_\beta(\mathbf{C}_\alpha - \mathbf{P}^\star)}{\mathbf{C}_\alpha - \mathbf{C}_\beta}, \ \mathbf{P}^\star = \begin{cases} P^L, \text{ if } (C_\alpha - C_\beta)(\gamma_\alpha - \gamma_\beta) > 0 \\ P^U, \text{ if } (C_\alpha - C_\beta)(\gamma_\alpha - \gamma_\beta) < 0 \end{cases}$$

Both Inputs and Directs: Intuition

$$\max_{\{x_i\},\{z_i\}} f = d \cdot D^U - \sum_{i \in T_X} \gamma_i \cdot x_i - \sum_{i \in T_Z} \gamma_i \cdot z_i$$

$$\begin{cases} \sum_{i \in T_X} x_i + \sum_{i \in T_Z} z_i = D^U \\ P^L \leq \left(\sum_{i \in T_X} C_i x_i + \sum_{i \in T_Z} C_i z_i \right) \Big/ \left(\sum_{i \in T_X} x_i + \sum_{i \in T_Z} z_i \right) \leq P^U \\ p = \sum_{i \in T_X} C_i x_i \Big/ \sum_{i \in T_X} x_i \\ 0 \leq x_i, \ \forall\, i \in T_X \qquad 0 \leq z_i, \ \forall\, i \in T_Z \end{cases}$$

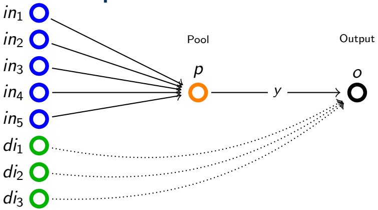


Calculate the breakpoints in strongly polynomial time (details: Optimization Online).

For a fixed p, the dominant active set is one of the following:

1. $\mathbf{x} = \{\mathbf{i}, \mathbf{j}\}$ delivering at output concentration $\mathbf{p}$ (equiv. to input $\rightarrow$ pool only).
2. $\mathbf{z} = \{\mathbf{k}, \mathbf{l}\}$ delivering at output concentration $\mathbf{P}^\star$ (equiv. to input $\rightarrow$ output only).
3. $\{\mathbf{x}, \mathbf{q}\} = \{\mathbf{i}, \mathbf{j}, \mathbf{q}\}$ delivering at output concentration $P(\mathbf{x}, \mathbf{q})$, where $\mathbf{x}$ is the blend of nodes $\mathbf{i}, \mathbf{j}$ with concentration $\mathbf{p}$.

Both Inputs and Directs: Example & Algorithm



$(C_{in_1} = 1, \gamma_{in_1} = 135)$, $(C_{in_2} = 2, \gamma_{in_2} = 105)$,
$(C_{in_3} = 3, \gamma_{in_3} = 90)$, $(C_{in_4} = 4, \gamma_{in_4} = 111)$,
$(C_{in_5} = 5, \gamma_{in_5} = 140)$;
$(C_{di_1} = 1.5, \gamma_{di_1} = 150)$, $(C_{di_2} = 8, \gamma_{di_2} = 115)$,
$(C_{di_3} = 4.5, \gamma_{di_3} = 120)$,
$D^U = 100$; $d = 150$; $P^L = 1$; $P^U = 3$;

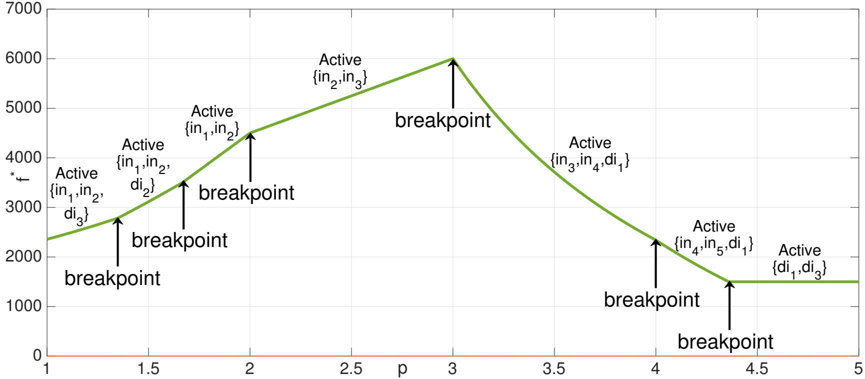


Figure 2: Optimal Objective Function $f^*(p)$ vs. pool concentration p

Both Inputs and Directs: Example & Algorithm

Over a Breakpoint (p) Interval I:

- If a direct-only active set z dominates over I, $f^*(p)$ is **constant** w.r.t. p over I.
- If an input-only active set x dominates over I, $f^*(p)$ is **linear** w.r.t. p over I.
- If a mixed (input and direct) active set m dominates over I, $f^*(p)$ is **monotone and either convex or concave** w.r.t. p over I.

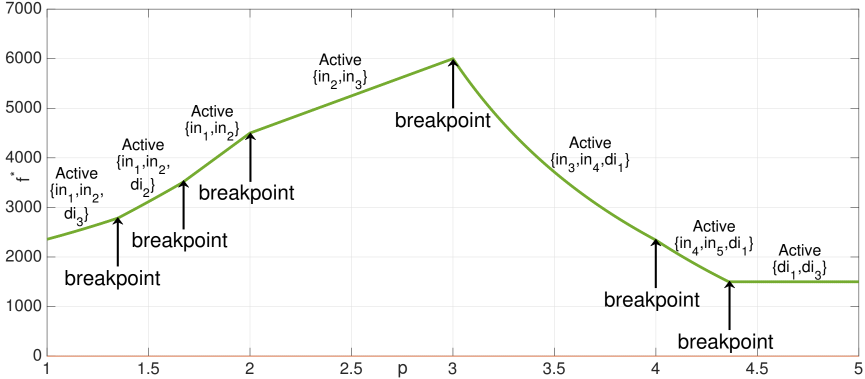


Figure 2: Optimal Objective Function $f^*(p)$ vs. pool concentration p

Both Inputs and Directs: Example & Algorithm

Finding the Global Solution

In strongly polynomial-time, find optimal objective $f^* = \max\left(\max_{p \in B} f^*(p), f_z^*\right)$, where B is the set of all breakpoints between dominant input-only and mixed active sets, and f_z^* is the optimal objective for the dominant direct-only active set.

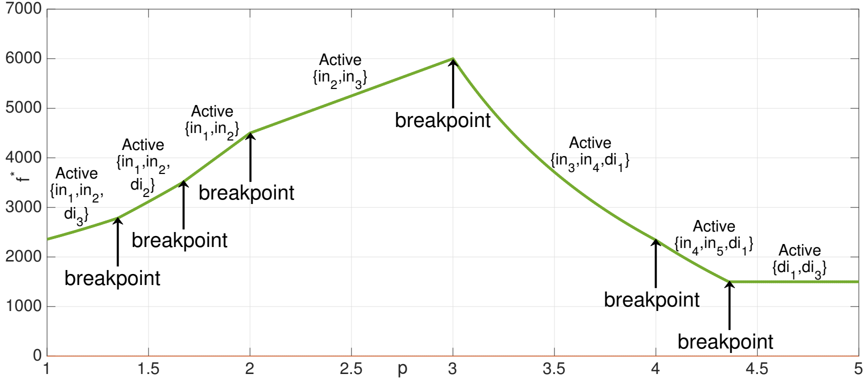


Figure 2: Optimal Objective Function $f^*(p)$ vs. pool concentration p

One Pool, Multiple Outputs: Problem

Objective $\max_{\{x_i\},\{y_j\},\{z_{ij}\},p} f = \sum_{j \in T_Y} d_j \cdot y_j + \sum_{(i,j) \in T_Z} d_j \cdot z_{ij} - \sum_{i \in T_X} \gamma_i \cdot x_i - \sum_{(i,j) \in T_Z} \gamma_i \cdot z_{ij}$

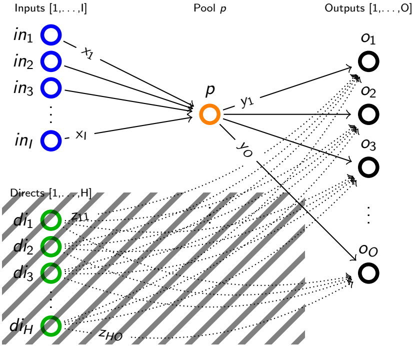
Product Demand $\left[y_j + \sum_{i:(i,j) \in T_Z} z_{ij} \le D_j^U \quad \forall\, j \right.$

Material Balance $\left[\sum_{i \in T_X} x_i - \sum_{j \in T_Y} y_j = 0 \right.$

Quality Balance $\left[\sum_{i \in T_X} C_i x_i = p \sum_{j \in T_Y} y_j \right.$

Product Quality $\left[p \cdot y_j + \sum_{i:(i,j) \in T_Z} C_i \cdot z_{ij} \begin{cases} \ge \mathbf{P_j^L} \left(\mathbf{y_j} + \sum_{\mathbf{i:(i,j) \in T_Z}} \mathbf{z_{ij}} \right) \\ \le P_j^U \left(y_j + \sum_{i:(i,j) \in T_Z} z_{ij} \right) \end{cases} \forall\, j \right.$

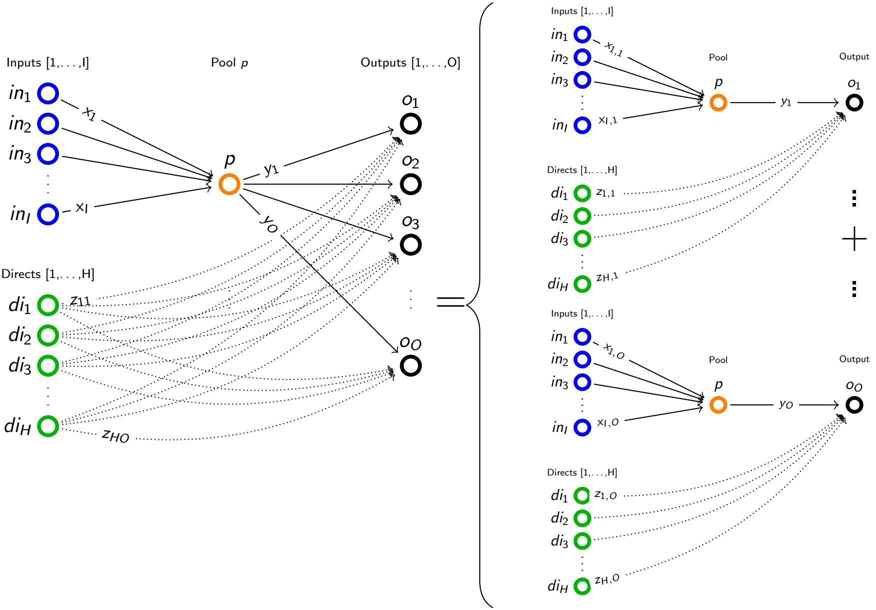
Bounds $\left[x_i,\ y_j,\ z_{ij} \ge 0\ \forall\ i,\ j \right.$



Known result

Can solve with a polynomial number of LPs of polynomial size [Haugland and Hendrix, 2016, Boland et al., 2016]

One Pool, Multiple Outputs: Additivity Intuition



One Pool, Multiple Outputs: Additivity over Outputs

$$\max_{\{x_i\},\{y_j\},\{z_{ij}\},p} f = \sum_{j\in T_Y} d_j\cdot y_j + \sum_{(i,j)\in T_Z} d_j\cdot z_{ij} - \sum_{i\in T_X}\gamma_i\cdot x_i - \sum_{(i,j)\in T_Z}\gamma_i\cdot z_{ij}$$

$$\left[\, y_j + \sum_{i:(i,j)\in T_Z} z_{ij} \le D_j^U \quad \forall\, j \right.$$

$$\left[\, \sum_{i\in T_X} x_i - \sum_{j\in T_Y} y_j = 0 \right.$$

$$\left[\, p\cdot y_j + \sum_{i:(i,j)\in T_Z} C_i\cdot z_{ij} \begin{cases} \ge P_j^L\left(y_j + \sum_{i:(i,j)\in T_Z} z_{ij}\right) \\ \le P_j^U\left(y_j + \sum_{i:(i,j)\in T_Z} z_{ij}\right) \end{cases} \quad \forall\, j \right.$$

$$\left[\, \sum_{i\in T_X} C_i x_i = p\sum_{j\in T_Y} y_j \right.$$

$$\left[\begin{array}{rccl} 0 \le & x_i & & \forall\, i\in T_X \\ 0 \le & y_j & & \forall\, j\in T_Y \\ 0 \le & z_{ij} & & \forall\, (i,j)\in T_Z \\ \min_i C_i \le & p & & \le \max_i C_i \end{array}\right.$$

$$\begin{cases} f(p) = \sum_{j\in T_Y} f_j(p) \\ \Leftrightarrow \max_{\{x_i\},\{y_j\},\{z_{ij}\},p} f = \\ = \max_p \left(\sum_{j\in T_Y} \max_{\{x_{ij}\},\{z_{ij}\}} f_j(p)\right), \\ x_i = \sum_{j\in T_Y} x_{ij} \quad \forall i\in T_X, \\ y_j = \sum_{i\in T_X} x_{ij} \quad \forall j\in T_Y, \\ p = \sum_{i\in T_X} C_i x_i \Big/ \sum_{j\in T_Y} y_j, \end{cases}$$

$$\Longleftrightarrow$$

For j given, $\forall j$:

$$\max_{\{x_{ij}\},\{z_{ij}\}} f_j(p) = d_j\cdot y_j + \sum_{i:(i,j)\in T_Z} d_j\cdot z_{ij} - \sum_{i\in T_X}\gamma_i\cdot x_{ij} - \sum_{i:(i,j)\in T_Z}\gamma_i\cdot z_{ij}$$

$$\left[\, y_j + \sum_{i:(i,j)\in T_Z} z_{ij} \le D_j^U \right.$$

$$\left[\, \sum_{i\in T_X} x_{ij} - y_j = 0 \right.$$

$$\left[\, p\cdot y_j + \sum_{i:(i,j)\in T_Z} C_i\cdot z_{ij} \begin{cases} \ge P_j^L\left(y_j + \sum_{i:(i,j)\in T_Z} z_{ij}\right) \\ \le P_j^U\left(y_j + \sum_{i:(i,j)\in T_Z} z_{ij}\right) \end{cases} \right.$$

$$\left[\, \sum_{i\in T_X} C_i x_{ij} = p\cdot y_j \right.$$

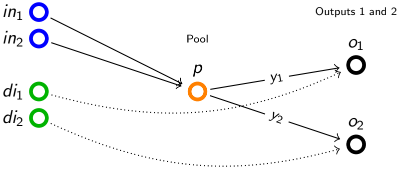
$$\left[\begin{array}{rccl} 0 \le & x_{ij} & & \forall\, i\in T_X \\ 0 \le & y_j & & \\ 0 \le & z_{ij} & & \forall\, i:(i,j)\in T_Z \\ \min_i C_i \le & p & & \le \max_i C_i \end{array}\right.$$

One Pool, Multiple Outputs: Decomposed Problem

$$\max_{\{x_i\},\{y_j\},\{z_{ij}\},p} f = \max_p \left(\sum_{j\in T_Y} \max_{\{x_{ij}\},\{z_{ij}\}} f_j(p) \right),$$

$$\begin{cases} x_i = \sum_{j\in T_Y} x_{ij} \quad \forall i \in T_X, \\ y_j = \sum_{i\in T_X} x_{ij} \quad \forall j \in T_Y, \\ p = \sum_{i\in T_X} C_i x_i \Big/ \sum_{j\in T_Y} y_j, \\ \forall j \in T_Y \quad \begin{cases} \max_{\{x_{ij}\},\{z_{ij}\}} f_j(p) = d_j \cdot D_j^U - \sum_{i\in T_X} \gamma_i \cdot x_{ij} - \sum_{i\in T_Z} \gamma_i \cdot z_{ij}, \\ \begin{cases} \sum_{i\in T_X} x_{ij} + \sum_{i\in T_Z} z_{ij} = D_j^U, \\ P_j^L \le \left(\sum_{i\in T_X} C_i x_{ij} + \sum_{i\in T_Z} C_i z_{ij} \right) \Big/ \left(\sum_{i\in T_X} x_{ij} + \sum_{i\in T_Z} z_{ij} \right) \le P_j^U, \\ p = \sum_{i\in T_X} C_i x_{ij} \Big/ \sum_{i\in T_X} x_{ij}, \\ 0 \le x_{ij},\ \forall\ i \in T_X \qquad 0 \le z_{ij},\ \forall\ i \in T_Z. \end{cases} \end{cases} \end{cases}$$

One Pool, Multiple Outputs: Numerical Example



$(C_{in_1} = 3, \gamma_{in_1} = 6)$, $(C_{in_2} = 1, \gamma_{in_2} = 13)$,
$(C_{di_1} = 1.5, \gamma_{di_1} = 12)$, $(C_{di_2} = 2.5, \gamma_{di_2} = 3)$,
$D_1^U = 200$; $D_2^U = 150$; $d_1 = 15$; $d_2 = 15$;
$P_1^L = 1$; $P_1^U = 2$; $P_2^L = 1$; $P_2^U = 2$;

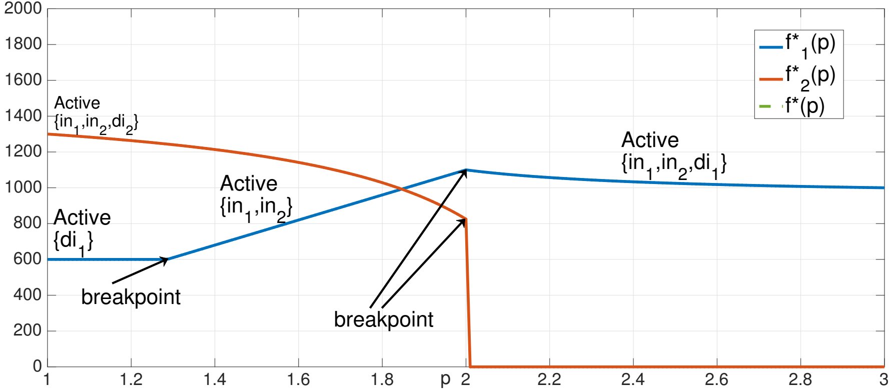
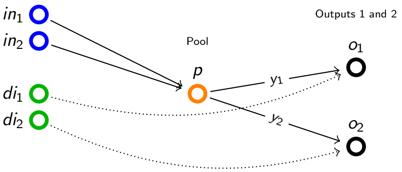


Figure 3: Optimal Objective Function $f^*(p)$ vs. pool concentration p

One Pool, Multiple Outputs: Numerical Example



$(C_{in_1} = 3, \gamma_{in_1} = 6),\ (C_{in_2} = 1, \gamma_{in_2} = 13),$
$(C_{di_1} = 1.5, \gamma_{di_1} = 12),\ (C_{di_2} = 2.5, \gamma_{di_2} = 3),$
$D_1^U = 200;\ D_2^U = 150;\ d_1 = 15;\ d_2 = 15;$
$P_1^L = 1;\ P_1^U = 2;\ P_2^L = 1;\ P_2^U = 2;$

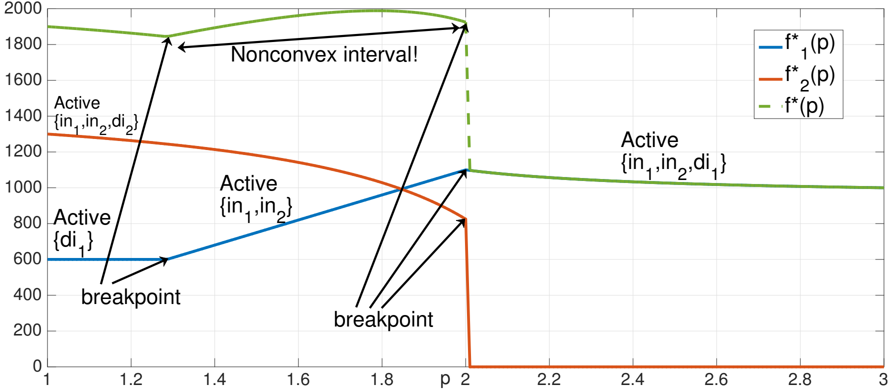


Figure 3: Optimal Objective Function $f^*(p)$ vs. pool concentration p

One Pool, Multiple Outputs: Algorithm

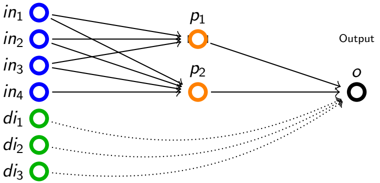
Overcoming Non-Convexity

- $f^*(p)$ can be non-convex, but we have its derivative over a breakpoint interval I (additivity over outputs):

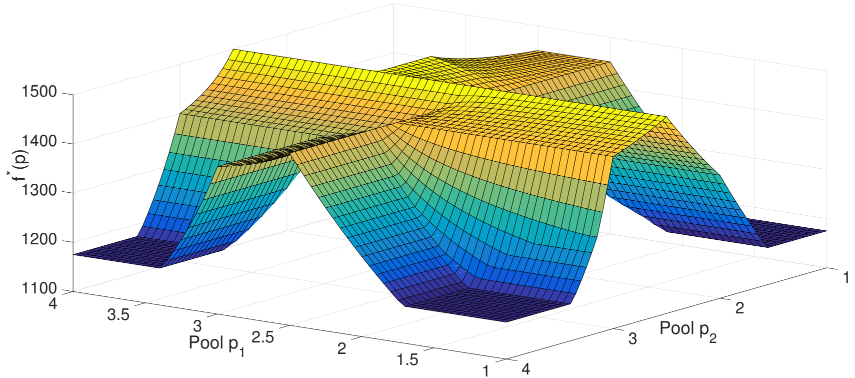
$$\frac{\partial f^*(p)}{\partial p} = \Bigg(\sum_{j \in T_{Y_m}} \frac{D_j^U (C_q^j - P^j(x,q))}{(C_i^j - C_k^j)(p - C_q^j)^2} \cdot \big(C_q^j(\gamma_k^j - \gamma_i^j) + C_j^j(\gamma_i^j - \gamma_q^j) + C_i^j(\gamma_q^j - \gamma_k^j) \big) \Bigg) + C,$$

  where C is a determined constant, j superscript is the output and T_{Y_m} are the outputs where a mixed active set dominates over I.
- $2|T_Y|$-bounded degree polynomial when derivative is $\partial f^*(p)/\partial p = 0$ has rational coefficients and can be solved in polynomial-time by *Lenstra-Lenstra-Lovacz* algorithm.

Two Pools, One Output: Numerical Example



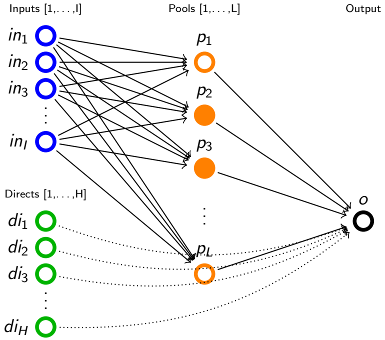
$(C_{in_1} = 3, \gamma_{in_1} = 6)$, $(C_{in_2} = 1, \gamma_{in_2} = 13)$,
$(C_{in_3} = 4, \gamma_{in_3} = 10)$, $(C_{in_4} = 2, \gamma_{in_4} = 9)$;
$(C_{di_1} = 2, \gamma_{di_1} = 10)$, $(C_{di_2} = 6, \gamma_{di_2} = 3)$,
$(C_{di_3} = 4, \gamma_{di_3} = 12)$,
$D^U = 200$; $d = 15$; $P^L = 1$; $P^U = 2.5$;



Multiple Pools: Two Pools, One Output - Why Important?

For a one output problem

- At any given p pool concentrations vector, only a maximum of two pools are active (have incoming and outgoing flow strictly non-zero) at the optimal solution of the problem.
- The L-pool problem considered can be solved by finding the maximum between the optimal objectives of all 2-pool sub-problems contained in a discrete set S, where $|S| = \binom{L}{2}$.



Other Extensions

For a multiple pools / multiple outputs problem

Since every output has a maximum of two active pools feeding into it at optimality, the following cases arise:

- If 2 outputs share 2 active pools: Additivity over outputs as in 1 pool case;
- If 2 outputs share no active pools: Separable problem over outputs;
- P / NP boundary when 2 outputs share only one active pool.

Full pooling problem

- Add back feed availability, pool capacity and lower product demand;
- Relax assumed profitability;
- Detect infeasibility;
- Multiple qualities.

References I

Alfaki and Haugland. A multi-commodity flow formulation for the generalized pooling problem. *J Glob Optim*, 56(3):917–937, 2013.

Beale, Coen, and Flowerdew. Separable programming applied to an ore purchasing problem. *J Roy Statistical Soc. Series C (Applied Statistics)*, 14(2/3):89–101, 1965.

Boland, Kalinowski, and Rigterink. A polynomially solvable case of the pooling problem. *J Glob Optim*, 2016. DOI 10.1007/s10898-016-0432-6.

Ceccon, Kouyialis, and Misener. Using functional programming to recognize named structure in an optimization problem: Application to pooling. *AIChE J*, 2016. ISSN 1547-5905. doi: 10.1002/aic.15308.

C D'Ambrosio, J Linderoth, and J Luedtke. *IPCO 2011*, chapter Valid Inequalities for the Pooling Problem with Binary Variables, pages 117–129. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

References II

Dey and Gupte. Analysis of MILP techniques for the pooling problem. *Oper Res*, 63(2):412–427, 2015.

Gupte, Ahmed, Dey, and Cheon. Relaxations and discretizations for the pooling problem. *J Glob Optim*, 2016. DOI 10.1007/s10898-016-0434-4.

Haugland. The computational complexity of the pooling problem. *J Glob Optim*, 64:199–215, 2016.

Haugland and Hendrix. Pooling problems with polynomial-time algorithms. *J Optim Theory Appl*, 2016. DOI 10.1007/s10957-016-0890-5.

Haverly. Studies of the behavior of recursion for the pooling problem. *ACM SIGMAP Bull.*, 25:19 – 28, 1978.