# Strong formulations for the pooling problem

Mohammed Alfaki

Department of Informatics, University of Bergen
Joint work with: Dag Haugland

TOGO10 - Global Optimization
Workshop, France, Toulouse
September 1, 2010

# Outline

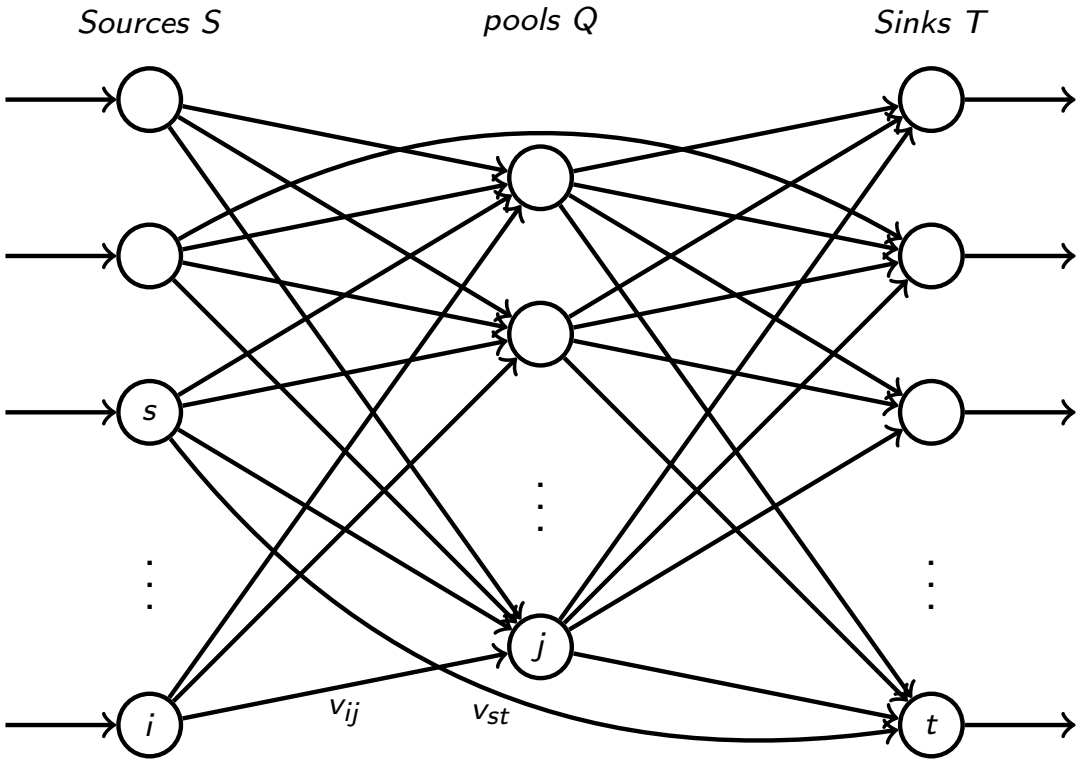


# Network topology

Given a directed acyclic graph $G = (N, A)$ with 3 sets of nodes

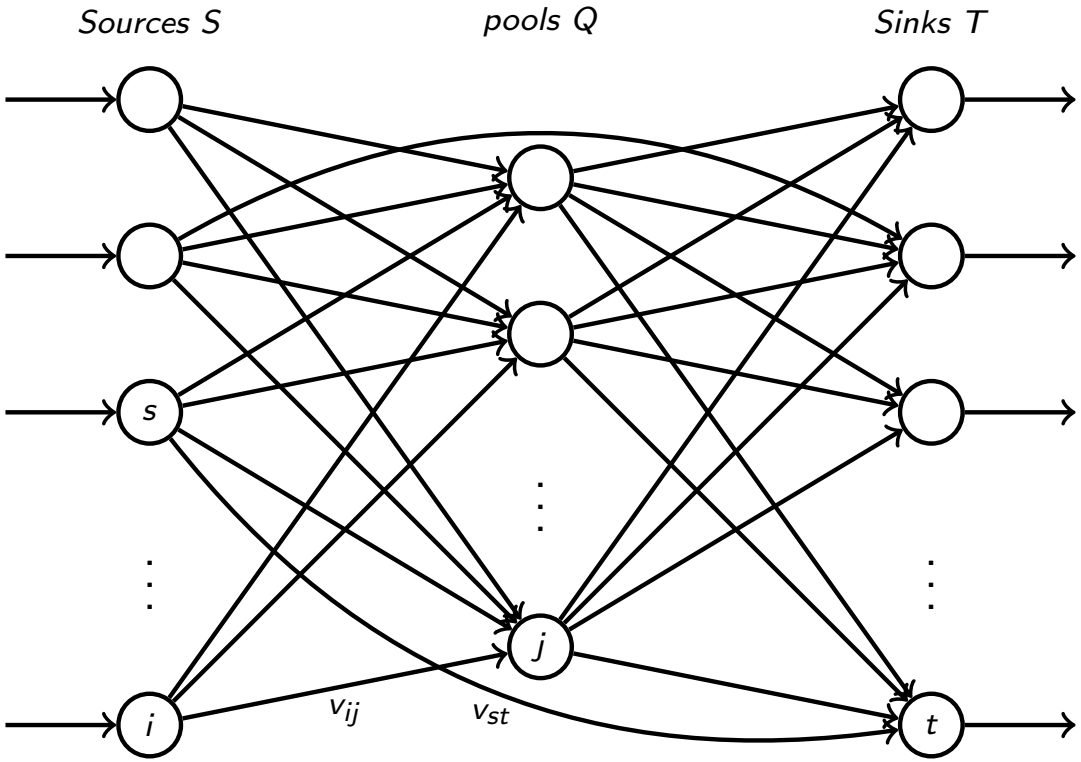
# Network topology

Given a directed acyclic graph $G = (N, A)$ with 3 sets of nodes



# Network topology

Given a directed acyclic graph $G = (N, A)$ with 3 sets of nodes




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

# Notations

## Define the parameters

- $K$ is a set of quality attributes

# Notations

## Define the parameters

- $K$ is a set of quality attributes
- flow capacity: $u_i$, $i \in N$

# Notations

## Define the parameters

- $K$ is a set of quality attributes
- flow capacity: $u_i$, $i \in N$
- unit cost: $c_{ij}$, $(i,j) \in A$

# Notations

## Define the parameters

- $K$ is a set of quality attributes
- flow capacity: $u_i$, $i \in N$
- unit cost: $c_{ij}$, $(i,j) \in A$
- quality bound: $q_t^k$, $t \in T, k \in K$

# Notations

## Define the parameters

- $K$ is a set of quality attributes
- flow capacity: $u_i$, $i \in N$
- unit cost: $c_{ij}$, $(i,j) \in A$
- quality bound: $q_t^k$, $t \in T, k \in K$
- quality parameter: $q_s^k$, $s \in S, k \in K$

# Notations

## Define the parameters

- $K$ is a set of quality attributes
- flow capacity: $u_i,\ i \in N$
- unit cost: $c_{ij},\ (i,j) \in A$
- quality bound: $q_t^k,\ t \in T, k \in K$
- quality parameter: $q_s^k,\ s \in S, k \in K$

## $\mathcal{F}(G, S, T, u)$ is the polytope

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

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

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

# Notations

## Define the parameters

- $K$ is a set of quality attributes
- flow capacity: $u_i,\ i \in N$
- unit cost: $c_{ij},\ (i,j) \in A$
- quality bound: $q_t^k,\ t \in T, k \in K$
- quality parameter: $q_s^k,\ s \in S, k \in K$

## $\mathcal{F}(G, S, T, u)$ is the polytope

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

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

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

$$\text{MCFP} = \min\{c^T v :\ v \in \mathcal{F}(G, S, T, u), 0 \leq v_{ij} \leq \max(u_i, u_j)\}$$

# The pooling problem

For any $v \in \mathcal{F}(G, S, T, u)$ associate a quality matrix $w \in \mathbb{R}^{N \times K}$:

$$w_i^k = \begin{cases} q_i^k & \text{if } i \in S \\ \dfrac{\sum_{j:(j,i)\in A} w_j^k v_{ji}}{\sum_{j:(j,i)\in A} v_{ji}} & \text{if } i \in N \setminus S \end{cases}$$

# The pooling problem

For any $v \in \mathcal{F}(G, S, T, u)$ associate a quality matrix $w \in \mathbb{R}^{N \times K}$:

$$w_i^k = \begin{cases} q_i^k & \text{if } i \in S \\ \dfrac{\sum_{j:(j,i)\in A} w_j^k v_{ji}}{\sum_{j:(j,i)\in A} v_{ji}} & \text{if } i \in N \setminus S \end{cases}$$

## Definition: The pooling problem

Find $v \in \mathcal{F}(G, S, T, u)$ with associated quality matrix $w \in \mathbb{R}^{N \times K}$ satisfying $w_t^k \leq q_t^k \ \forall t \in T, k \in K$ such that $c^T v$ is minimized.

# The pooling problem

For any $v \in \mathcal{F}(G, S, T, u)$ associate a quality matrix $w \in \mathbb{R}^{N \times K}$:

$$w_i^k = \begin{cases} q_i^k & \text{if } i \in S \\ \dfrac{\sum_{j:(j,i)\in A} w_j^k v_{ji}}{\sum_{j:(j,i)\in A} v_{ji}} & \text{if } i \in N \setminus S \end{cases}$$

**Definition: The pooling problem**

Find $v \in \mathcal{F}(G, S, T, u)$ with associated quality matrix $w \in \mathbb{R}^{N \times K}$ satisfying $w_t^k \leq q_t^k \ \forall t \in T, k \in K$ such that $c^T v$ is minimized.

The definition resembles the P-formulation

# Maximum 3-dimensional matching

**Definition**

Given 3 disjoint finite sets $I_1$, $I_2$ & $I_3$, and a set $E \subseteq I_1 \times I_2 \times I_3$, find $M \subseteq E$ of maximum cardinality s.t $\forall\ i \in I_1 \cup I_2 \cup I_3$ there is at most one $(i_1, i_2, i_3) \in M$ s.t $i \in \{i_1, i_2, i_3\}$

# Folklore - the pooling problem is $\mathcal{NP}$-hard

Reducing max 3D matching to the pooling problem

- $S \sim I_1$

# Folklore - the pooling problem is $\mathcal{NP}$-hard

Reducing max 3D matching to the pooling problem

- $S \sim I_1$
- $Q \sim I_2$

# Folklore - the pooling problem is $\mathcal{NP}$-hard

Reducing max 3D matching to the pooling problem

- $S \sim I_1$
- $Q \sim I_2$
- $T \sim (i_1, i_3)$ : $i_1$ and $i_3$ appear jointly in at least one triple in $E$

# Folklore - the pooling problem is $\mathcal{NP}$-hard

Reducing max 3D matching to the pooling problem

- $S \sim I_1$
- $Q \sim I_2$
- $T \sim (i_1, i_3)$ : $i_1$ and $i_3$ appear jointly in at least one triple in $E$
- $K \sim \{k_i : i \in I_1\}$

# Folklore - the pooling problem is $\mathcal{NP}$-hard

Reducing max 3D matching to the pooling problem

- $S \sim I_1$
- $Q \sim I_2$
- $T \sim (i_1, i_3)$ : $i_1$ and $i_3$ appear jointly in at least one triple in $E$
- $K \sim \{k_i : i \in I_1\}$
- $A = \{(i_1, i_2) : \exists\, (i_1, i_2, i_3) \in E\} \cup \{(i_2, (i_1, i_3)) : (i_1, i_2, i_3) \in E\}$

# Folklore - the pooling problem is $\mathcal{NP}$-hard

Reducing max 3D matching to the pooling problem

- $S \sim I_1$
- $Q \sim I_2$
- $T \sim (i_1, i_3)$ : $i_1$ and $i_3$ appear jointly in at least one triple in $E$
- $K \sim \{k_i : i \in I_1\}$
- $A = \{(i_1, i_2) : \exists\, (i_1, i_2, i_3) \in E\} \cup \{(i_2, (i_1, i_3)) : (i_1, i_2, i_3) \in E\}$
- $u_i = 1,\ \forall\, i \in N$ and $c_{ij} = -1,\ \forall\, (i, j) \in A$

# Folklore - the pooling problem is $\mathcal{NP}$-hard

Reducing max 3D matching to the pooling problem

- $S \sim I_1$
- $Q \sim I_2$
- $T \sim (i_1, i_3)$ : $i_1$ and $i_3$ appear jointly in at least one triple in $E$
- $K \sim \{k_i : i \in I_1\}$
- $A = \{(i_1, i_2) : \exists\, (i_1, i_2, i_3) \in E\} \cup \{(i_2, (i_1, i_3)) : (i_1, i_2, i_3) \in E\}$
- $u_i = 1,\ \forall\ i \in N$ and $c_{ij} = -1,\ \forall\ (i,j) \in A$
- $\forall\ t = (i_1, i_3)$, let $q_t^k = \begin{cases} 0 & k = k_{i_1} \\ 1 & k \neq k_{i_1} \end{cases}$, $q_{i_1}^k = \begin{cases} 0 & k = k_{i_1} \\ 1 & k \neq k_{i_1} \end{cases}$

# Consequence

Through out the proof we assume that $|K|$ is changing with the problem size

# Consequence

Through out the proof we assume that $|K|$ is changing with the problem size

## Open question

Is the pooling problem $\mathcal{NP}$-hard for fixed $|K|$?

# PQ-formulation

- Q-formulation (Ben-Tal et. al 94): introduce the proportion

$$y_i^s = \frac{v_{si}}{\sum_{t:(i,t)\in A} v_{it}}, \; s \in S, \; i \in Q$$

# PQ-formulation

- Q-formulation (Ben-Tal et. al 94): introduce the proportion

$$y_i^s = \frac{v_{si}}{\sum_{t:(i,t)\in A} v_{it}}, \ s \in S, \ i \in Q$$

- PQ-formulation (Sahinidis & Tawarmalani 05): add the RLT cuts to the Q-formulation

$$\sum_{s\in S} y_i^s v_{it} = v_{it}, \ i \in Q, \ t \in T$$

$$\sum_{t\in T} y_i^s v_{it} \leq u_i y_i^s, \ s \in S, \ i \in Q$$

# PQ-formulation

- Q-formulation (Ben-Tal et. al 94): introduce the proportion

$$y_i^s = \frac{v_{si}}{\sum_{t:(i,t)\in A} v_{it}}, \ s \in S, \ i \in Q$$

- PQ-formulation (Sahinidis & Tawarmalani 05): add the RLT cuts to the Q-formulation

$$\sum_{s\in S} y_i^s v_{it} = v_{it}, \ i \in Q, \ t \in T$$

$$\sum_{t\in T} y_i^s v_{it} \leq u_i y_i^s, \ s \in S, \ i \in Q$$

- PQ-formulation dominates both the P- and the Q-formulation

# TP-formulation

- analogous to the proportion $y_i^s$, define the terminal proportion

$$y_i^t = \frac{v_{it}}{\sum_{s:(s,i)\in A} v_{si}}, \ t \in T, \ i \in Q$$

# TP-formulation

- analogous to the proportion $y_i^s$, define the terminal proportion

$$y_i^t = \frac{v_{it}}{\sum_{s:(s,i)\in A} v_{si}}, \ t \in T, \ i \in Q$$

- adding the cuts

$$\sum_{t\in T} y_i^t v_{si} = v_{si}, \ s \in S, \ i \in Q$$

$$\sum_{s\in S} y_i^t v_{si} \leq u_i y_i^t, \ t \in T, \ i \in Q$$

# TP-formulation

- analogous to the proportion $y_i^s$, define the terminal proportion

$$y_i^t = \frac{v_{it}}{\sum_{s:(s,i)\in A} v_{si}}, \ t \in T, \ i \in Q$$

- adding the cuts

$$\sum_{t\in T} y_i^t v_{si} = v_{si}, \ s \in S, \ i \in Q$$

$$\sum_{s\in S} y_i^t v_{si} \leq u_i y_i^t, \ t \in T, \ i \in Q$$

- comparison to the PQ-formulation

# TP-formulation

- analogous to the proportion $y_i^s$, define the terminal proportion

$$y_i^t = \frac{v_{it}}{\sum_{s:(s,i)\in A} v_{si}}, \ t \in T, \ i \in Q$$

- adding the cuts

$$\sum_{t\in T} y_i^t v_{si} = v_{si}, \ s \in S, \ i \in Q$$

$$\sum_{s\in S} y_i^t v_{si} \leq u_i y_i^t, \ t \in T, \ i \in Q$$

- comparison to the PQ-formulation
  1. they do not have equal strength

# TP-formulation

- analogous to the proportion $y_i^s$, define the terminal proportion

$$y_i^t = \frac{v_{it}}{\sum_{s:(s,i)\in A} v_{si}}, \ t \in T, \ i \in Q$$

- adding the cuts

$$\sum_{t\in T} y_i^t v_{si} = v_{si}, \ s \in S, \ i \in Q$$

$$\sum_{s\in S} y_i^t v_{si} \leq u_i y_i^t, \ t \in T, \ i \in Q$$

- comparison to the PQ-formulation
  1. they do not have equal strength
  2. none of them dominates the other

# STP-formulation

- combine the source and terminal proportions

# STP-formulation

- combine the source and terminal proportions
- define $x_{sit}$ as the flow on the path $(s, i, t)$ in $G$

# STP-formulation

- combine the source and terminal proportions
- define $x_{sit}$ as the flow on the path $(s, i, t)$ in $G$
- observe that

$$x_{sit} = y_i^s v_{it} \tag{1}$$

$$x_{sit} = y_i^t v_{si} \tag{2}$$

# STP-formulation

- combine the source and terminal proportions
- define $x_{sit}$ as the flow on the path $(s, i, t)$ in $G$
- observe that

$$x_{sit} = y_i^s v_{it} \tag{1}$$

$$x_{sit} = y_i^t v_{si} \tag{2}$$

- this model dominates both the PQ- and the TP-formulation

# STP-formulation

- combine the source and terminal proportions
- define $x_{sit}$ as the flow on the path $(s, i, t)$ in $G$
- observe that

$$x_{sit} = y_i^s v_{it} \tag{1}$$
$$x_{sit} = y_i^t v_{si} \tag{2}$$

- this model dominates both the PQ- and the TP-formulation
- given a rectangle for each bilinear term the linearization achieved by replacing (1) and (2) by

# STP-formulation

- combine the source and terminal proportions
- define $x_{sit}$ as the flow on the path $(s, i, t)$ in $G$
- observe that

$$x_{sit} = y_i^s v_{it} \tag{1}$$
$$x_{sit} = y_i^t v_{si} \tag{2}$$

- this model dominates both the PQ- and the TP-formulation
- given a rectangle for each bilinear term the linearization achieved by replacing (1) and (2) by
  1. Vex$(y_i^s v_{it})$, and Cav$(y_i^s v_{it})$

# STP-formulation

- combine the source and terminal proportions
- define $x_{sit}$ as the flow on the path $(s, i, t)$ in $G$
- observe that

$$x_{sit} = y_i^s v_{it} \tag{1}$$
$$x_{sit} = y_i^t v_{si} \tag{2}$$

- this model dominates both the PQ- and the TP-formulation
- given a rectangle for each bilinear term the linearization achieved by replacing (1) and (2) by
  1. Vex$(y_i^s v_{it})$, and Cav$(y_i^s v_{it})$
  2. Vex$(y_i^t v_{si})$, and Cav$(y_i^t v_{si})$

# Branching strategy

A procedure for choosing the branching pair in the STP-formulation

1. for each $(s, i, t)$

$$\text{violation} = \min\{v^* y^* - \text{Vex}(vy), \text{Cav}(vy) - v^* y^*\}$$

2. find the maximum violation pair in $\{y_i^s v_{it}\}$
3. find the maximum violation pair in $\{y_i^t v_{si}\}$
4. choose the pair with minimum violation from (2) and (3)

# Comparing the relaxations

Red value $\Longrightarrow$ better lower bounds

| Problem | Objective function value | | | Global Solution |
|---|---|---|---|---|
| | PQ-relaxation | TP-relaxation | STP-relaxation | |
| Adhya1 | −840.27 | −856.25 | −840.27 | −549.80 |
| Adhya2 | −574.78 | −574.78 | −574.78 | −549.80 |
| Adhya3 | −574.78 | −574.78 | −574.78 | −561.05 |
| Adhya4 | −961.93 | −967.43 | −961.93 | −877.65 |
| Bental4 | −550.00 | −541.67 | −541.67 | −450.00 |
| Bental5 | −3500.00 | −3500.00 | −3500.00 | −3500.27 |
| Foulds2 | −1100.00 | −1100.00 | −1100.00 | −1100.00 |
| Foulds3 | −8.00 | −8.00 | −8.00 | −8.00 |
| Foulds4 | −8.00 | −8.00 | −8.00 | −8.00 |
| Foulds5 | −8.00 | −8.00 | −8.00 | −8.00 |
| Haverly1 | −500.00 | −500.00 | −500.00 | −400.00 |
| Haverly2 | −1000.00 | −1000.00 | −1000.00 | −600.00 |
| Haverly3 | −800.00 | −875.00 | −800.00 | −750.00 |
| Rt2 | −6034.87 | −5528.00 | −5528.25 | −4391.83 |

# Computational experiments with BARON

- formulate the PQ-, TP-, and STP-formulation on GAMS

# Computational experiments with BARON

- formulate the PQ-, TP-, and STP-formulation on GAMS
- BARON as global solver

# Computational experiments with BARON

- formulate the PQ-, TP-, and STP-formulation on GAMS
- BARON as global solver
- neglect the RLT cuts in the local search procedure

# Computational experiments with BARON

- formulate the PQ-, TP-, and STP-formulation on GAMS
- BARON as global solver
- neglect the RLT cuts in the local search procedure
  - apply the `RELAXATION_ONLY_EQUATIONS` option in BARON

# Computational experiments with BARON

| Problem | PQ-formulation | | TP-formulation | | STP-formulation | |
|---|---|---|---|---|---|---|
| | #nodes | time(sec) | #nodes | time(sec) | #nodes | time(sec) |
| Adhya1 | 21 | 0.20 | 39 | 0.61 | 7 | 0.37 |
| Adhya2 | 33 | 0.19 | 15 | 0.30 | 1 | 0.27 |
| Adhya3 | 31 | 0.38 | 37 | 0.61 | 19 | 0.82 |
| Adhya4 | 1 | 0.18 | 17 | 0.27 | 1 | 0.30 |
| Bental4 | 1 | 0.02 | 1 | 0.01 | 1 | 0.02 |
| Bental5 | 0 | 0.03 | 0 | 0.10 | 0 | 0.17 |
| Foulds2 | 0 | 0.02 | 0 | 0.01 | 0 | 0.02 |
| Foulds3 | 0 | 0.84 | 0 | 0.36 | 0 | 8.95 |
| Foulds4 | 0 | 1.18 | 0 | 0.50 | 0 | 1.57 |
| Foulds5 | 0 | 0.30 | 0 | 0.42 | 0 | 1.64 |
| Haverly1 | 1 | 0.02 | 1 | 0.02 | 1 | 0.02 |
| Haverly2 | 5 | 0.02 | 1 | 0.01 | 1 | 0.03 |
| Haverly3 | 1 | 0.02 | 1 | 0.01 | 1 | 0.02 |
| Rt2 | 13 | 0.14 | 1 | 0.10 | 1 | 0.19 |

0 nodes means that the problem solved during the preprocessing

# Large problems

Randomly generated problems with

- specified $|S|$, $|Q|$, $|T|$, and $|K|$

# Large problems

Randomly generated problems with

- specified $|S|$, $|Q|$, $|T|$, and $|K|$
- a given probability of the network density

# Large problems

Randomly generated problems with

- specified $|S|$, $|Q|$, $|T|$, and $|K|$
- a given probability of the network density
- all other parameters generated within specified intervals

# Large problems

Randomly generated problems with

- specified $|S|$, $|Q|$, $|T|$, and $|K|$
- a given probability of the network density
- all other parameters generated within specified intervals

| problem | problem sets size | | | | probability |
|---|---|---|---|---|---|
| | $\|S\|$ | $\|Q\|$ | $\|T\|$ | $\|K\|$ | |
| problem1 | 25 | 15 | 20 | 10 | 0.60 |
| problem2 | 40 | 25 | 35 | 15 | 0.52 |
| problem3 | 60 | 35 | 50 | 20 | 0.30 |

# Computational results–large problems

We have coded a simple branch-and-bound algorithm in C++

- optimality-based bound tightening (OBBT) in the root

# Computational results–large problems

We have coded a simple branch-and-bound algorithm in C++

- optimality-based bound tightening (OBBT) in the root
  - using OBBT to provide upper bound

# Computational results–large problems

We have coded a simple branch-and-bound algorithm in C++

- optimality-based bound tightening (OBBT) in the root
  - using OBBT to provide upper bound
- implement STP-formulation branching strategy

# Computational results–large problems

We have coded a simple branch-and-bound algorithm in C++

- optimality-based bound tightening (OBBT) in the root
  - using OBBT to provide upper bound
- implement STP-formulation branching strategy
- CPLEX as lp solver

# Computational results–large problems

We have coded a simple branch-and-bound algorithm in C++

- optimality-based bound tightening (OBBT) in the root
  - using OBBT to provide upper bound
- implement STP-formulation branching strategy
- CPLEX as lp solver

| Instance | PQ-formulation | | | | STP-formulation | | | |
|---|---|---|---|---|---|---|---|---|
| | BARON | | Our Code | | BARON | | Our Code | |
| | # node | time(sec) | # node | time(sec) | # node | time(sec) | # node | time(sec) |
| problem1 | 0 | 739.20 | 298 | 273.47 | – | > 2 days | 151 | 830.28 |
| problem2 | – | > 2 days | 532 | 4273.84 | – | > 2 days | 303 | 41100.15 |
| problem3 | – | > 2 days | 714 | 41387.13 | – | > 2 days | 640 | 162526.54 |

# Summary

- the pooling problem is $\mathcal{NP}$-hard (a formal proof)

# Summary

- the pooling problem is $\mathcal{NP}$-hard (a formal proof)
- derived a new formulation that proved to be stronger than PQ-formulation

# Summary

- the pooling problem is $\mathcal{NP}$-hard (a formal proof)
- derived a new formulation that proved to be stronger than PQ-formulation
- a branching strategy suite with the STP-formulation

Thank you.

# proof illustration

consider any $t = (i_1, i_3) \in T$, $t$ can be supplied along the path from $i_1$ via some pool $i_2 \in Q$ to $t$

# proof illustration

consider any $t = (i_1, i_3) \in T$, $t$ can be supplied along the path from $i_1$ via some pool $i_2 \in Q$ to $t$, since

$$w_{i_2}^{k_{i_1}} = 0 \quad \left( w_{i_2}^{k_{i_1}} \leq q_t^{k_{i_1}} \right) \quad \Longrightarrow$$

# proof illustration

consider any $t = (i_1, i_3) \in T$, $t$ can be supplied along the path from $i_1$ via some pool $i_2 \in Q$ to $t$, since

$$w_{i_2}^{k_{i_1}} = 0 \quad \left( w_{i_2}^{k_{i_1}} \leq q_t^{k_{i_1}} \right) \quad \Longrightarrow$$

$$\sum_{j:(j,i_2)\in A} w_j^{k_{i_1}} v_{ji_2} = 0 \quad (u_{i_2} = 1) \quad \Longrightarrow v_{ji_2} = \begin{cases} 0 & \text{if } j \neq i_1 \\ 1 & \text{if } j = i_1 \end{cases}$$

# proof illustration

consider any $t = (i_1, i_3) \in T$, $t$ can be supplied along the path from $i_1$ via some pool $i_2 \in Q$ to $t$, since

$$w_{i_2}^{k_{i_1}} = 0 \quad \left( w_{i_2}^{k_{i_1}} \leq q_t^{k_{i_1}} \right) \quad \Longrightarrow$$

$$\sum_{j:(j,i_2)\in A} w_j^{k_{i_1}} v_{ji_2} = 0 \quad (u_{i_2} = 1) \quad \Longrightarrow v_{ji_2} = \begin{cases} 0 & \text{if } j \neq i_1 \\ 1 & \text{if } j = i_1 \end{cases}$$

then the path $(i_1, i_2, t)$ is the only one