Robust Network Design with Several Traffic Scenarios

Models and Algorithms

Eduardo Álvarez-Miranda[1] Valentina Cacchiani[1] Tim Dorneth[3]
Michael Jünger[3] Frauke Liers[2] Andrea Lodi[1] Tiziano Parriani[1]
Daniel R. Schmidt[3]

[1]DEIS, Università di Bologna

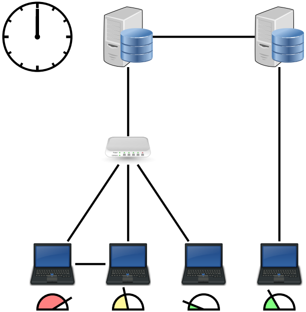
[2]Department Mathematik, Universität Erlangen-Nürnberg

[3]Institut für Informatik, Universität zu Köln

Financial support is acknowledged from the Ateneo Italo-Tedesco VIGONI programme and the DFG

Outline

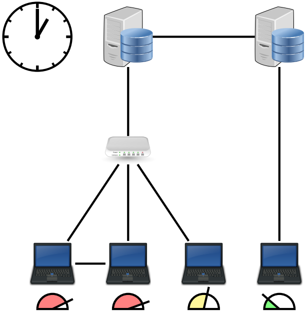
Motivation: A simple model for RND



Model [Buchheim, Liers, Sanità 2011]

- servers with identical data ("supply")
- users request data ("demand")
- all servers may answer user requests
- approximate demands by samples during the day
- use samples to design minimum cost network

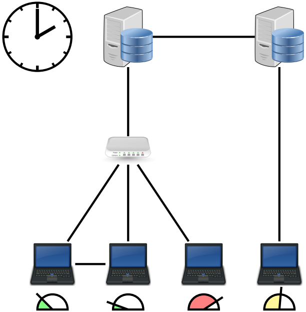
Motivation: A simple model for RND



Model [Buchheim, Liers, Sanità 2011]

- servers with identical data ("supply")
- users request data ("demand")
- all servers may answer user requests
- approximate demands by samples during the day
- use samples to design minimum cost network

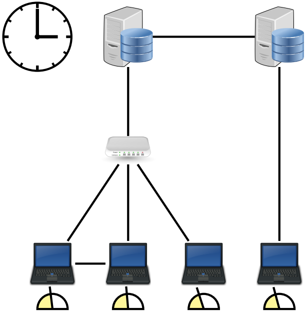
Motivation: A simple model for RND



Model [Buchheim, Liers, Sanità 2011]

- servers with identical data ("supply")
- users request data ("demand")
- all servers may answer user requests
- approximate demands by samples during the day
- use samples to design minimum cost network

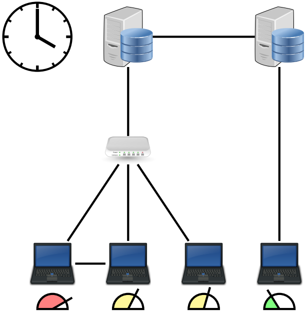
Motivation: A simple model for RND



Model [Buchheim, Liers, Sanità 2011]

- servers with identical data (“supply”)
- users request data (“demand”)
- all servers may answer user requests
- approximate demands by samples during the day
- use samples to design minimum cost network

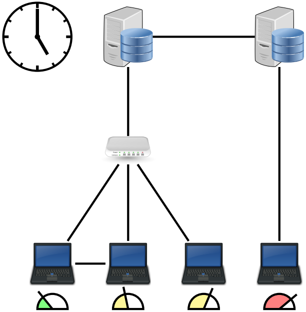
Motivation: A simple model for RND



Model [Buchheim, Liers, Sanità 2011]

- servers with identical data (“supply”)
- users request data (“demand”)
- all servers may answer user requests
- approximate demands by samples during the day
- use samples to design minimum cost network

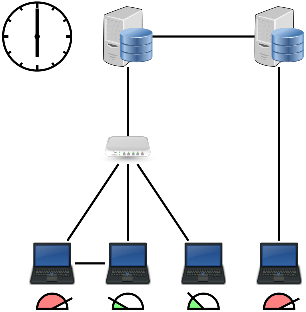
Motivation: A simple model for RND



Model [Buchheim, Liers, Sanità 2011]

- servers with identical data ("supply")
- users request data ("demand")
- all servers may answer user requests
- approximate demands by samples during the day
- use samples to design minimum cost network

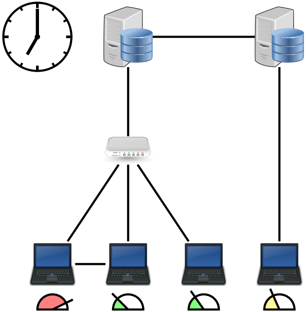
Motivation: A simple model for RND



Model [Buchheim, Liers, Sanità 2011]

- servers with identical data (“supply”)
- users request data (“demand”)
- all servers may answer user requests
- approximate demands by samples during the day
- use samples to design minimum cost network

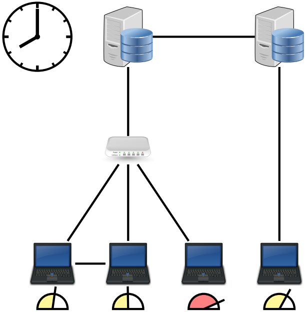
Motivation: A simple model for RND



Model [Buchheim, Liers, Sanità 2011]

- servers with identical data ("supply")
- users request data ("demand")
- all servers may answer user requests
- approximate demands by samples during the day
- use samples to design minimum cost network

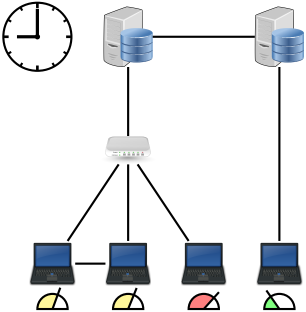
Motivation: A simple model for RND



Model [Buchheim, Liers, Sanità 2011]

- servers with identical data ("supply")
- users request data ("demand")
- all servers may answer user requests
- approximate demands by samples during the day
- use samples to design minimum cost network

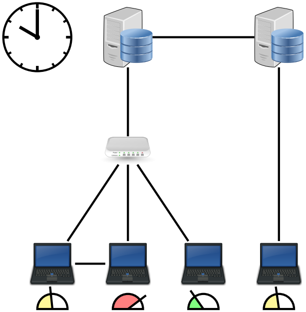
Motivation: A simple model for RND



Model [Buchheim, Liers, Sanità 2011]

- servers with identical data (“supply”)
- users request data (“demand”)
- all servers may answer user requests
- approximate demands by samples during the day
- use samples to design minimum cost network

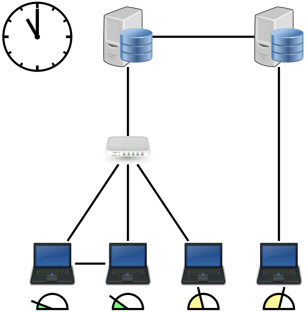
Motivation: A simple model for RND



Model [Buchheim, Liers, Sanità 2011]

- servers with identical data (“supply”)
- users request data (“demand”)
- all servers may answer user requests
- approximate demands by samples during the day
- use samples to design minimum cost network

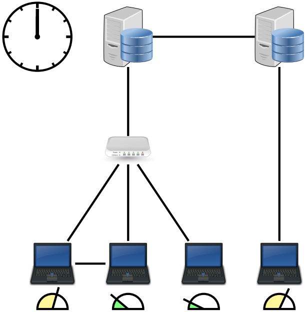
Motivation: A simple model for RND



Model [Buchheim, Liers, Sanità 2011]

- servers with identical data (“supply”)
- users request data (“demand”)
- all servers may answer user requests
- approximate demands by samples during the day
- use samples to design minimum cost network

Motivation: A simple model for RND



Model [Buchheim, Liers, Sanità 2011]

- servers with identical data ("supply")
- users request data ("demand")
- all servers may answer user requests
- approximate demands by samples during the day
- use samples to design minimum cost network

Single Commodity Robust Network Design

Given:

- undirected graph $G = (V, E)$
- cost vector $c : E \to \mathbb{Z}_{\geq 0}$
- K integer balance vectors $b_1, \ldots, b_K : V \to \mathbb{Z}$ ("scenarios")

Single Commodity Robust Network Design

Given:

- undirected graph $G = (V, E)$
- cost vector $c : E \to \mathbb{Z}_{\geq 0}$
- K integer balance vectors $b_1, \ldots, b_K : V \to \mathbb{Z}$ ("scenarios")

Task:

- find integer capacities $u : E \to \mathbb{Z}_{\geq 0}$
- s.t. there is directed b_i-flow f^i w.r.t. u for all $i = 1, \ldots, K$.

$$
\begin{aligned}
&(1) \qquad f^i_{u,v} + f^i_{v,u} \leq u_{u,v} \quad \text{for all } \{u, v\} \in E \\
&(2) \qquad \sum_{u \in \delta(v)} (f^i_{u,v} - f^i_{v,u}) = b^i_v \quad \text{for all } v \in V
\end{aligned}
$$

- minimize $c^T u$

Flow based IP-formulation

$$\min \sum_{\{i,j\}\in E} c_{i,j} u_{i,j}$$

$$f_{i,j}^q + f_{i,j}^q \le u_{i,j} \qquad \text{for all } \{i,j\} \in E,\ q = 1,\ldots,K$$

$$\sum_{j\in\delta(i)} (f_{i,j}^q - f_{j,i}^q) = b_i^q \qquad \text{for all } i \in V,\ q = 1,\ldots,K$$

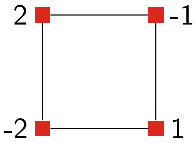
$$u_{i,j} \in \mathbb{Z}_{\ge 0} \qquad \text{for all } \{i,j\} \in E$$

$$f_{i,j}^q, f_{j,i}^q \in \mathbb{Z}_{\ge 0} \qquad \text{for all } \{i,j\} \in E,\ q = 1,\ldots,K$$

Example (unit edge costs)

Scenario 1

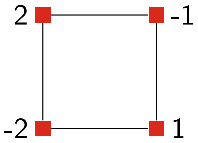
$c \equiv 1$



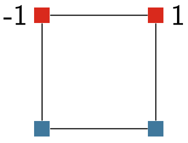
Example (unit edge costs)

$c \equiv 1$

Scenario 1



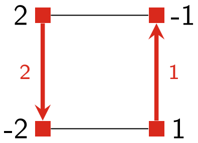
Scenario 2



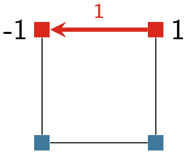
Example (unit edge costs)

Scenario 1

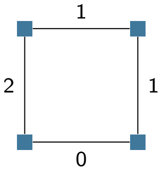
$c \equiv 1$



Scenario 2



Feasible solution:



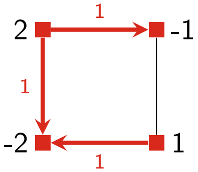
cost: 4

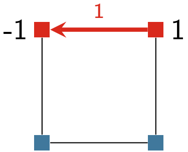
Example (unit edge costs)

Scenario 1

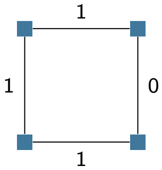
Scenario 2

$c \equiv 1$





Feasible solution:



cost: 3

Heuristic: Improvement Phase

Large Neighborhood Search

- improve feasible solution u^*
- for some $T \geq 0$, search T-neighborhood of u^*

➥ shift up to T units of capacity

Heuristic: Improvement Phase

Large Neighborhood Search

- improve feasible solution u^*
- for some $T \geq 0$, search T-neighborhood of u^*

➥ shift up to T units of capacity

Algorithm

1. Construct feasible solution u^*
2. Remove all edges e with $u^* = 0$
3. Solve IP on reduced graph $G = (V, \overline{E})$

Improvement phase IP

$$\min \sum_{\{i,j\} \in \overline{E}} c_{ij} u_{ij}$$

$$\text{s.t.} \quad \sum_{j \in \delta(i)} (f_{ij}^q - f_{ji}^q) = b_i^q \qquad \forall i \in V, q = 1, \dots, K$$

$$f_{ij}^q + f_{ji}^q \leq u_{ij} \qquad \forall \{i,j\} \in \overline{E}, q = 1, \dots, K$$

$$u_{ij} \leq u_{ij}^* + w_{ij} \quad \forall \{i,j\} \in \overline{E}$$

$$\sum_{\{i,j\} \in \overline{E}} w_{ij} \leq T$$

$$u \in \mathbb{Z}_{\geq 0}^{\overline{E}}, f \geq 0, w \geq 0$$

Improvement phase IP

$$\min \sum_{\{i,j\}\in\overline{E}} c_{ij} u_{ij}$$

$$\text{s.t.} \quad \sum_{j\in\delta(i)} (f_{ij}^q - f_{ji}^q) = b_i^q \qquad \forall i \in V, q = 1, \ldots, K$$

$$f_{ij}^q + f_{ji}^q \leq u_{ij} \qquad \forall \{i,j\} \in \overline{E}, q = 1, \ldots, K$$

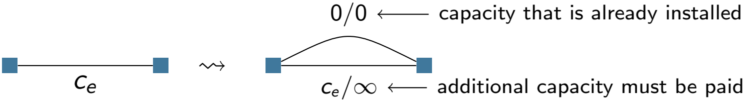
$$u_{ij} \leq u_{ij}^* + w_{ij} \quad \forall \{i,j\} \in \overline{E}$$

$$\sum_{\{i,j\}\in\overline{E}} w_{ij} \leq T$$

$$u \in \mathbb{Z}_{\geq 0}^{\overline{E}}, f \geq 0, w \geq 0$$

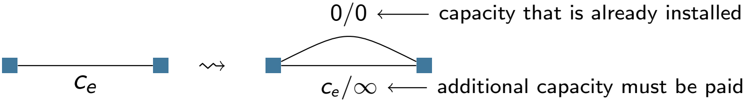
Heuristic: Constructive Phase

1. insert auxilliary edge (for all edges $e \in E$)



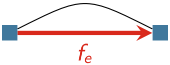
Heuristic: Constructive Phase

1. insert auxilliary edge (for all edges $e \in E$)

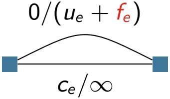


2. For $q = 1, \ldots, K$:

   1. compute MinCost flow for q-th scenario

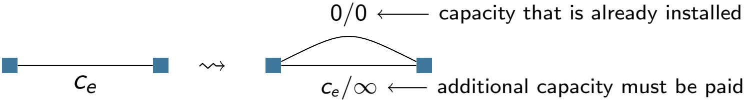


   2. update capacities for subsequent scenarios



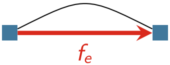
Heuristic: Constructive Phase

1. insert auxilliary edge (for all edges $e \in E$)

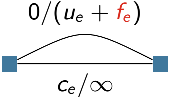


2. For $q = 1, \ldots, K$: Caution: The order matters!

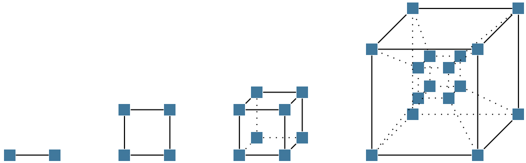
   1. compute MinCost flow for q-th scenario



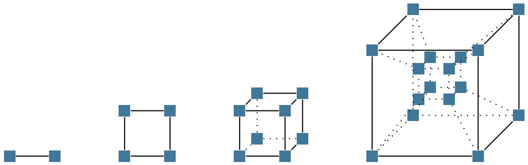
   2. update capacities for subsequent scenarios



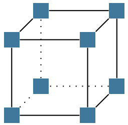
Class of difficult instances: hypercubes



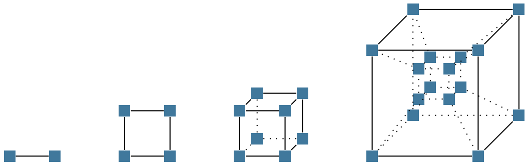
Class of difficult instances: hypercubes



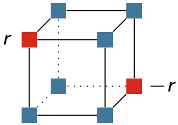
- Scenarios: Opposite nodes get balance of $\pm r$



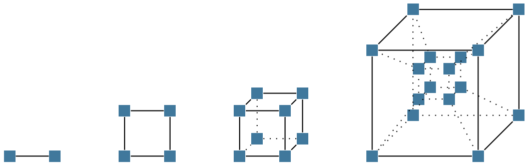
Class of difficult instances: hypercubes



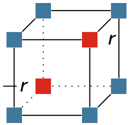
- Scenarios: Opposite nodes get balance of $\pm r$



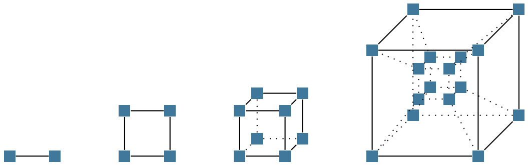
Class of difficult instances: hypercubes



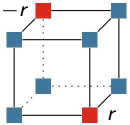
- Scenarios: Opposite nodes get balance of $\pm r$



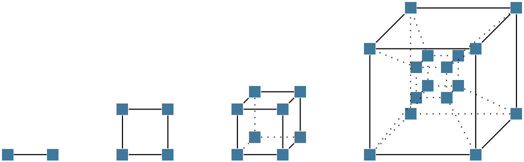
Class of difficult instances: hypercubes



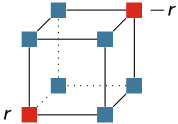
- Scenarios: Opposite nodes get balance of $\pm r$



Class of difficult instances: hypercubes



- Scenarios: Opposite nodes get balance of $\pm r$



Integrality gap on hypercubes

Ratio of best integer and best fractional solution

$$\text{GAP}(I) := \frac{OPT_{\text{int}}(I)}{OPT_{\text{frac}}(I)}$$

We can prove...

- $r = 1$: $GAP(I)$ converges to 2 as $d \to \infty$
- $r > 1$: $GAP(I)$ is 1 for all $d > 2$

$r = 1$: $GAP(I)$ converges to 2 as $d \to \infty$

Observe

$OPT_{frac} = r \cdot 2^{d-1}$

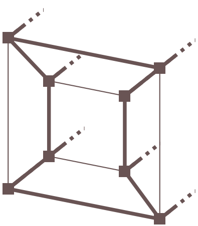
Analyze support of opt. int. solution

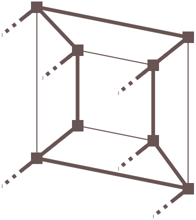
- Any connected component contains $\geq 2d$ nodes
- $\Rightarrow$ at most $2^d/(2d)$ connected components
- $\Rightarrow$ solution contains at least $2^d - 2^d/(2d)$ edges
- $GAP(I, d) \geq 2 - 1/d$

$r > 1$: $GAP(I)$ is 1 for all $d > 2$

Building large demands from smaller ones

- $r = 2$ and $d > 1$: costs $2^d = 2 \cdot 2^{d-1}$ (Hamiltonian cycle)
- $r = 3$ and $d > 2$: costs $3 \cdot 2^{d-1}$





- Glueing solutions: costs $r \cdot 2^{d-1}$

Computational Results

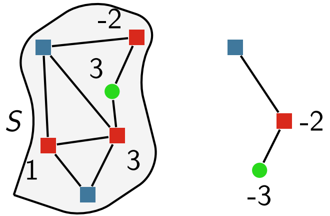
- instances: hypercubes, random r in each scenario
- Cplex 12.3, CS2 code by Goldberg for CP
- Intel(R) Core(TM) i7 CPU @ 1.73 GHz, 64 bit,
- time limit of 7200 sec.
- average values

Computational Results

- instances: hypercubes, random r in each scenario
- Cplex 12.3, CS2 code by Goldberg for CP
- Intel(R) Core(TM) i7 CPU @ 1.73 GHz, 64 bit,
- time limit of 7200 sec.
- average values

	CP		RND_{init}		$T = \infty$		$T = 25$	
dim.	time	%gap	time	%gap	time	%gap	time	%gap
6d	0.06s	16.10	1533.59s	0.25	21.82s	1.02	16.29	1.02
7d	0.24s	25.78	7200.00s	10.70	3003.73s	2.13	448.77s	4.02

New IP-formulation: Cut-Set-Inequalities

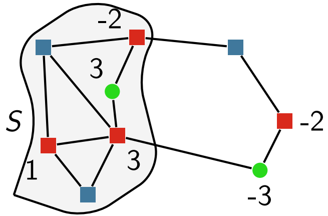


$$\min \sum_{\{u,v\} \in E} c_{u,v} u_{u,v}$$

$$\sum_{\{u,v\} \in \delta(S)} u_{u,v} \geq \max_{q=1,\dots,K} \left| \sum_{v \in S} b_v^q \right| \quad \text{for all } S \subseteq V$$

$$u_{u,v} \in \mathbb{Z}_{\geq 0} \quad \text{for all } \{u,v\} \in E$$

New IP-formulation: Cut-Set-Inequalities

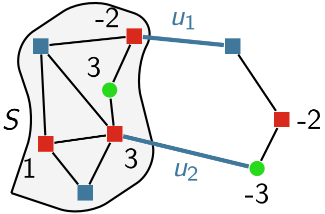


$$\min \sum_{\{u,v\}\in E} c_{u,v} u_{u,v}$$

$$\sum_{\{u,v\}\in\delta(S)} u_{u,v} \geq \max_{q=1,\ldots,K} \left| \sum_{v\in S} b_v^q \right| \quad \text{for all } S \subseteq V$$

$$u_{u,v} \in \mathbb{Z}_{\geq 0} \quad \text{for all } \{u,v\} \in E$$

New IP-formulation: Cut-Set-Inequalities

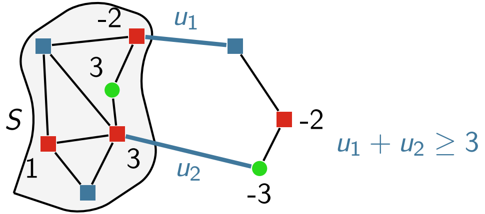


$$\min \sum_{\{u,v\}\in E} c_{u,v} u_{u,v}$$

$$\sum_{\{u,v\}\in\delta(S)} u_{u,v} \geq \max_{q=1,\dots,K} \left| \sum_{v\in S} b_v^q \right| \quad \text{for all } S \subseteq V$$

$$u_{u,v} \in \mathbb{Z}_{\geq 0} \quad \text{for all } \{u,v\} \in E$$

New IP-formulation: Cut-Set-Inequalities



$$\min \sum_{\{u,v\} \in E} c_{u,v} u_{u,v}$$

$$\sum_{\{u,v\} \in \delta(S)} u_{u,v} \geq \max_{q=1,\dots,K} \left| \sum_{v \in S} b_v^q \right| \quad \text{for all } S \subseteq V$$

$$u_{u,v} \in \mathbb{Z}_{\geq 0} \quad \text{for all } \{u,v\} \in E$$

Results: Branch-and-Cut

- Intel(R) Xeon(TM) 5680 @3.33Ghz CPU, 64bit
- time limit of 7200s
- CPLEX 12.1, ABACUS 3.2 (beta)

Results: Branch-and-Cut

- Intel(R) Xeon(TM) 5680 @3.33Ghz CPU, 64bit
- time limit of 7200s
- CPLEX 12.1, ABACUS 3.2 (beta)

Instances

- *d*-dimensional hypercubes
- terminal placement as before, random demands

Results: Branch-and-Cut

- Intel(R) Xeon(TM) 5680 @3.33Ghz CPU, 64bit
- time limit of 7200s
- CPLEX 12.1, ABACUS 3.2 (beta)

Instances

- d-dimensional hypercubes
- terminal placement as before, random demands

Results

- 2d, 3d, 4d: Solved to optimality in < 1s
- 5d: 7/10 solved to optimality, 3.0% avg. gap
- 6d: 2/10 solved to optimality, 4.7% avg. gap
- 7d: 9/10 with no bound, 1/10 with 14.7% gap

Results: Branch-and-Cut

- Intel(R) Xeon(TM) 5680 @3.33Ghz CPU, 64bit
- time limit of 7200s
- CPLEX 12.1, ABACUS 3.2 (beta)

Results: Branch-and-Cut

- Intel(R) Xeon(TM) 5680 @3.33Ghz CPU, 64bit
- time limit of 7200s
- CPLEX 12.1, ABACUS 3.2 (beta)

Instances

- based on [Altin, Amaldi, Belotti and Pinar 2007]
- used by [Buchheim, Liers, Sanità 2011]

Results: Branch-and-Cut

- Intel(R) Xeon(TM) 5680 @3.33Ghz CPU, 64bit
- time limit of 7200s
- CPLEX 12.1, ABACUS 3.2 (beta)

Instances

- based on [Altin, Amaldi, Belotti and Pinar 2007]
- used by [Buchheim, Liers, Sanità 2011]

Comparison to flow formulation based B&C

- [BLS 2011]: $\approx$ 94% solved to optimality
- Our approach: $\approx$ 75% solved to optimality

The end.