May MIP Techniques help to solve MINLPs?

A. Martin, jointly with B. Geißler, A. Morsi, L. Schewe
FAU Erlangen – Nürnberg, EDOM

June 2nd, 2014
Mixed-Integer Nonlinear Programming 2014
Carnegie Mellon University, Pittsburgh, PA

EDOM
Economics · Discrete
Optimization · Mathematics

FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Mixed-Integer Nonlinear Programs (MINLPs)

$$
\begin{array}{lll}
\min & \mathbf{c}^T\mathbf{x} & \\
\text{s.t.} & g_i(\mathbf{x}) = b_i & \text{for } i = 1, \ldots, k_1, \\
& h_i(\mathbf{x}) \geq b_i & \text{for } i = 1, \ldots, k_2, \\
& \mathbf{l} \leq \mathbf{x} \leq \mathbf{u}, & \mathbf{l}, \mathbf{u} \in \mathbb{R}^d, \\
& \mathbf{x} \in \mathbb{R}^{d-p} \times \mathbb{Z}^p, &
\end{array}
$$

Many Applications:

- Chemical-, electrical- and civil engineering
- Finance management
- ...
- Water and gas network optimization

Motivation

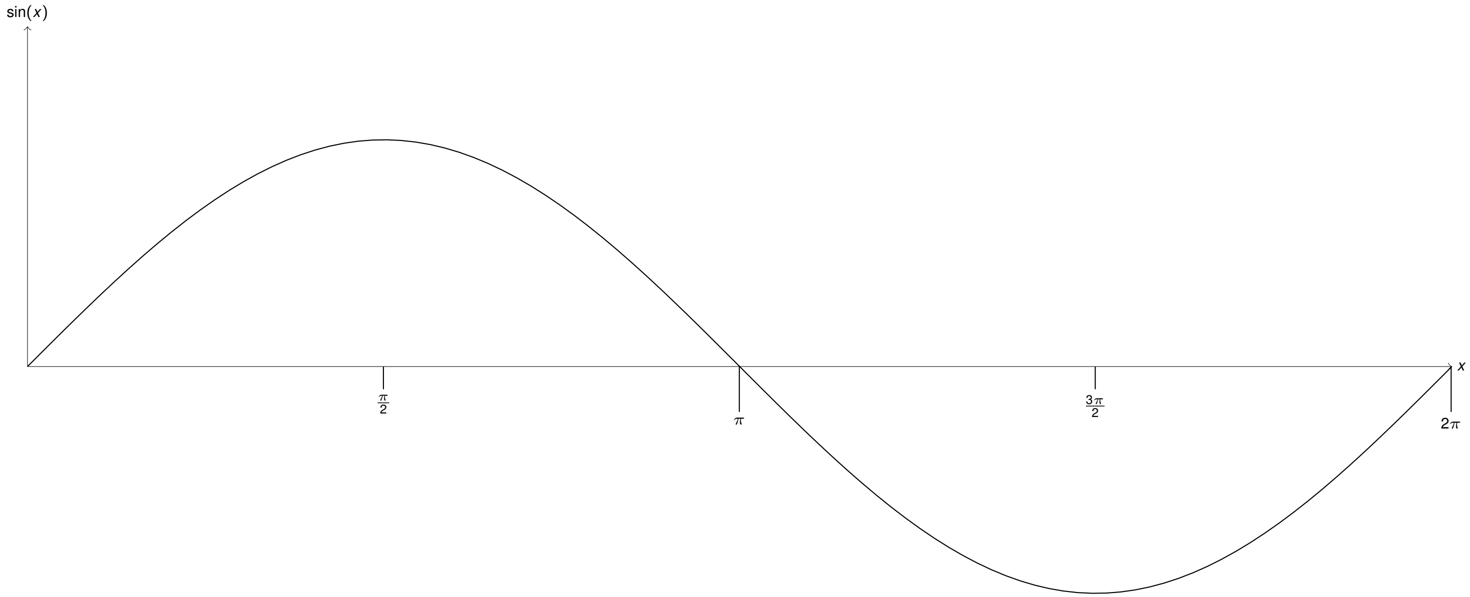
- Such problems are typically solved by outer approximation and spatial branching, cf.
  - Baron (Tawarmalani, Sahinidis 2005)
  - Couenne (Belotti, Lee, Liberti, Margot, Wächter 2009)
  - SCIP (Vigerske 2013)
  - alphaECP (Westerlund, Lindquist 2003)
  - Bonmin (Bonami, Biegler, Conn, Cornuejols, Grossmann, Laird, Lee, Lodi, Margot, Sawaya, Wächter 2005)
  - ...
- In case of nonconvex (MI)NLP spatial branching is unavoidable
- Lots of branching experiences for (M)IPs
- Polyhedral combinatorics help to avoid parts of branching

Motivation

Idea: For a given (MI)NLP and maximal absolute constraint violation $\epsilon > 0$ construct a MIP such that

1. the feasible set of the (MI)NLP is contained in the feasible set of the MIP
2. any point which is feasible to the MIP violates no constraint of the MINLP by more than ϵ

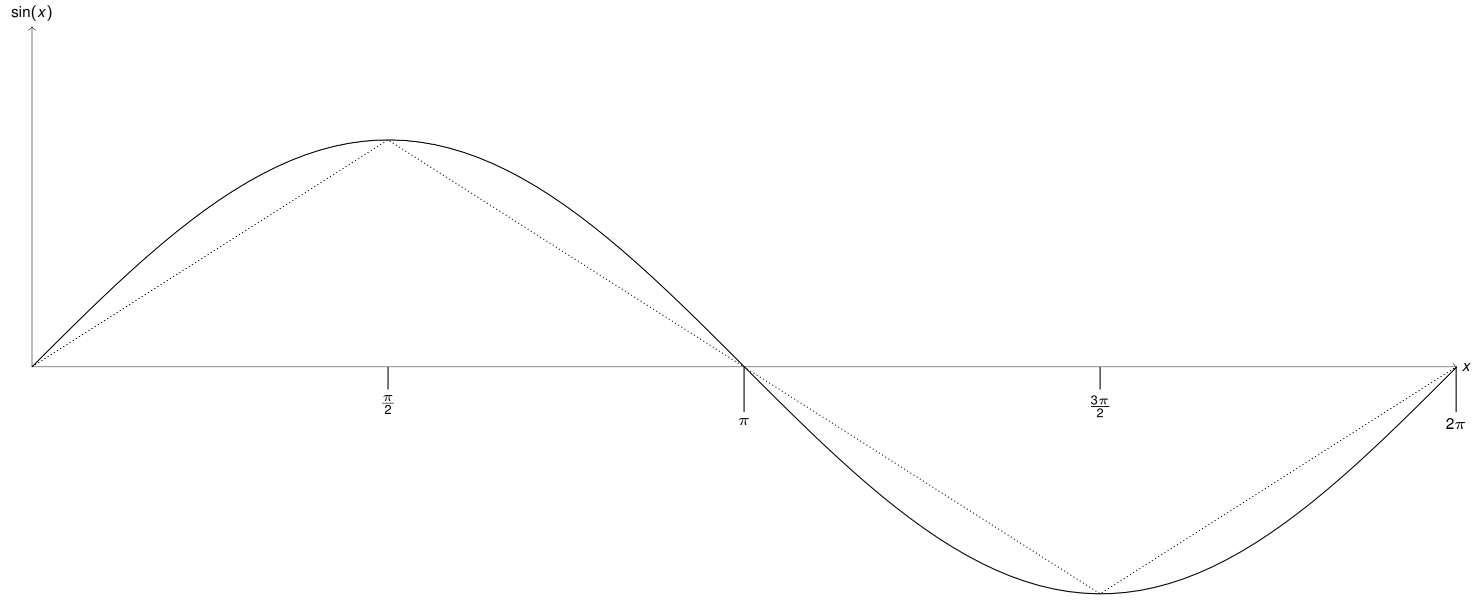
MIP-relaxations of MINLPs



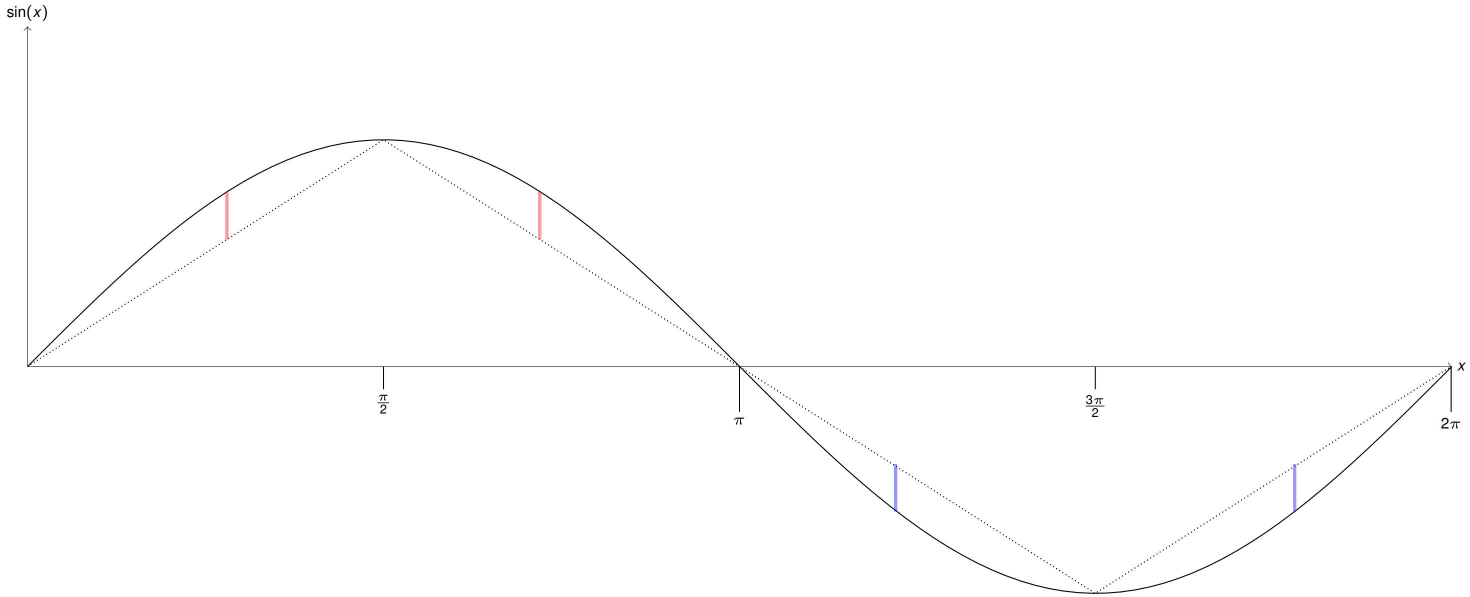
EDOM
Economics · Discrete Optimization · Mathematics

FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

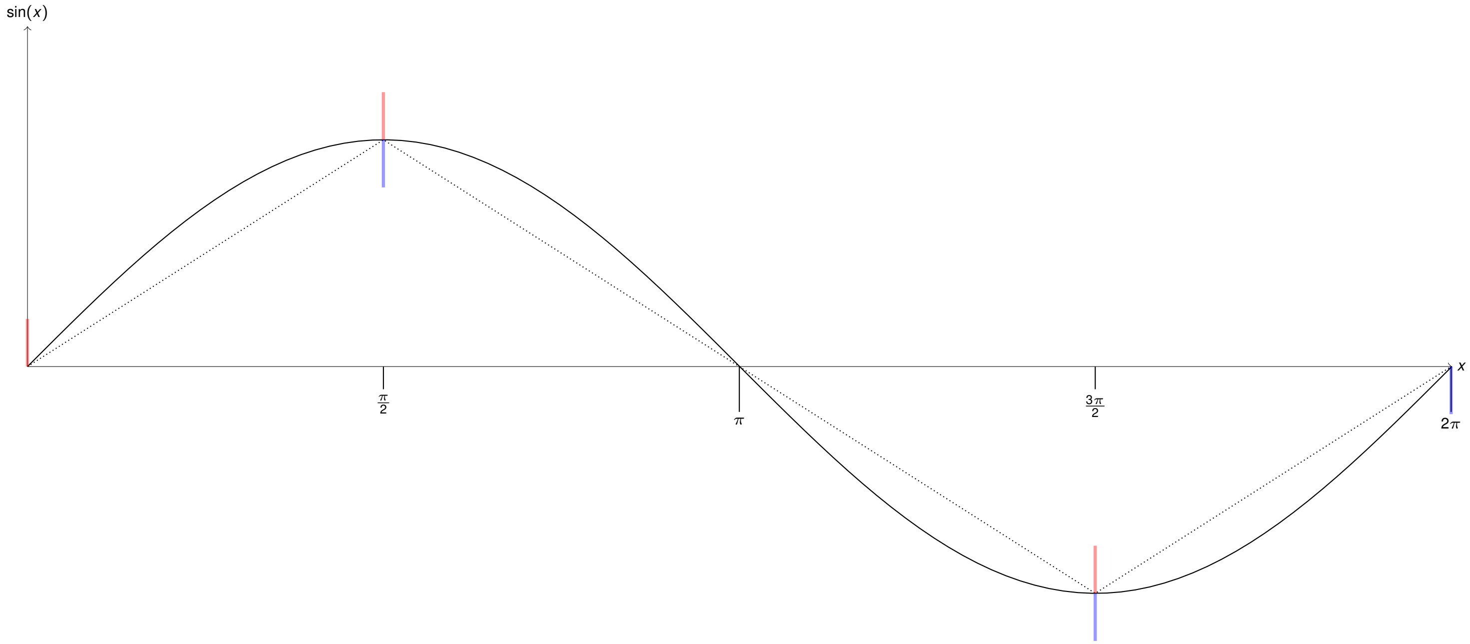
MIP-relaxations of MINLPs



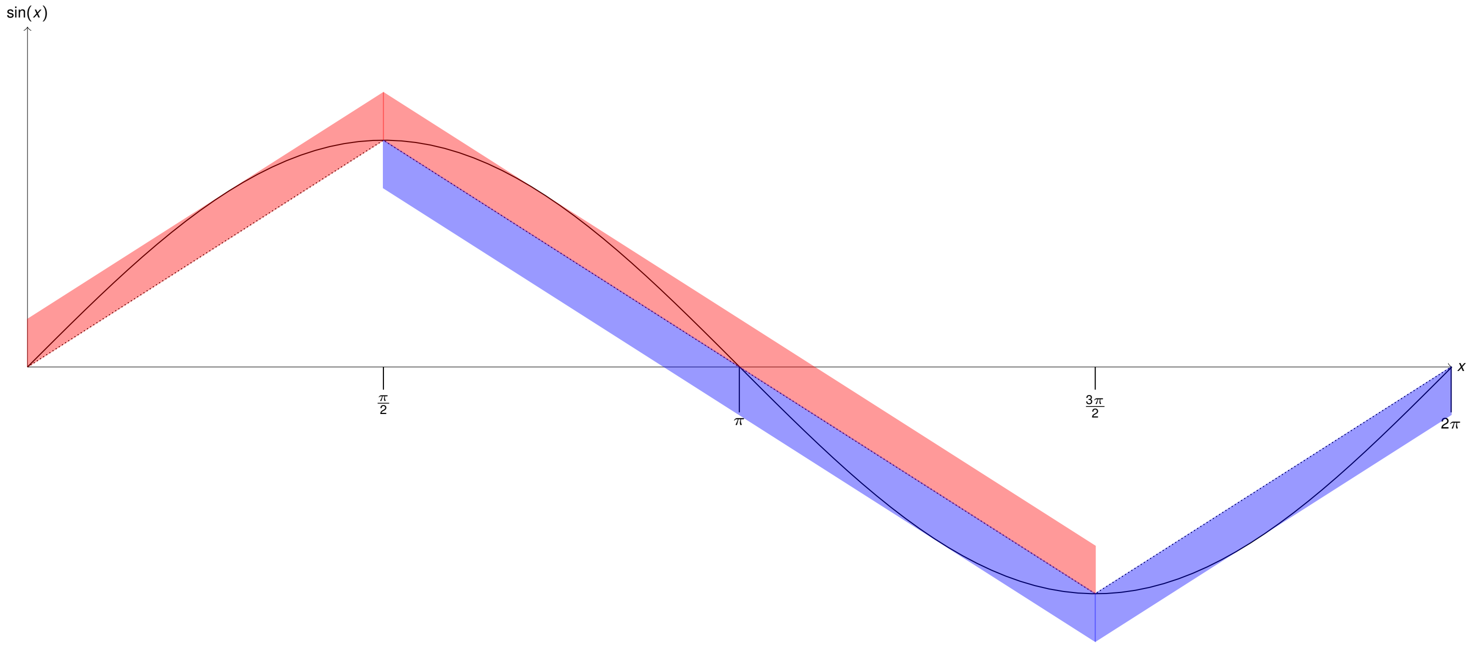
MIP-relaxations of MINLPs



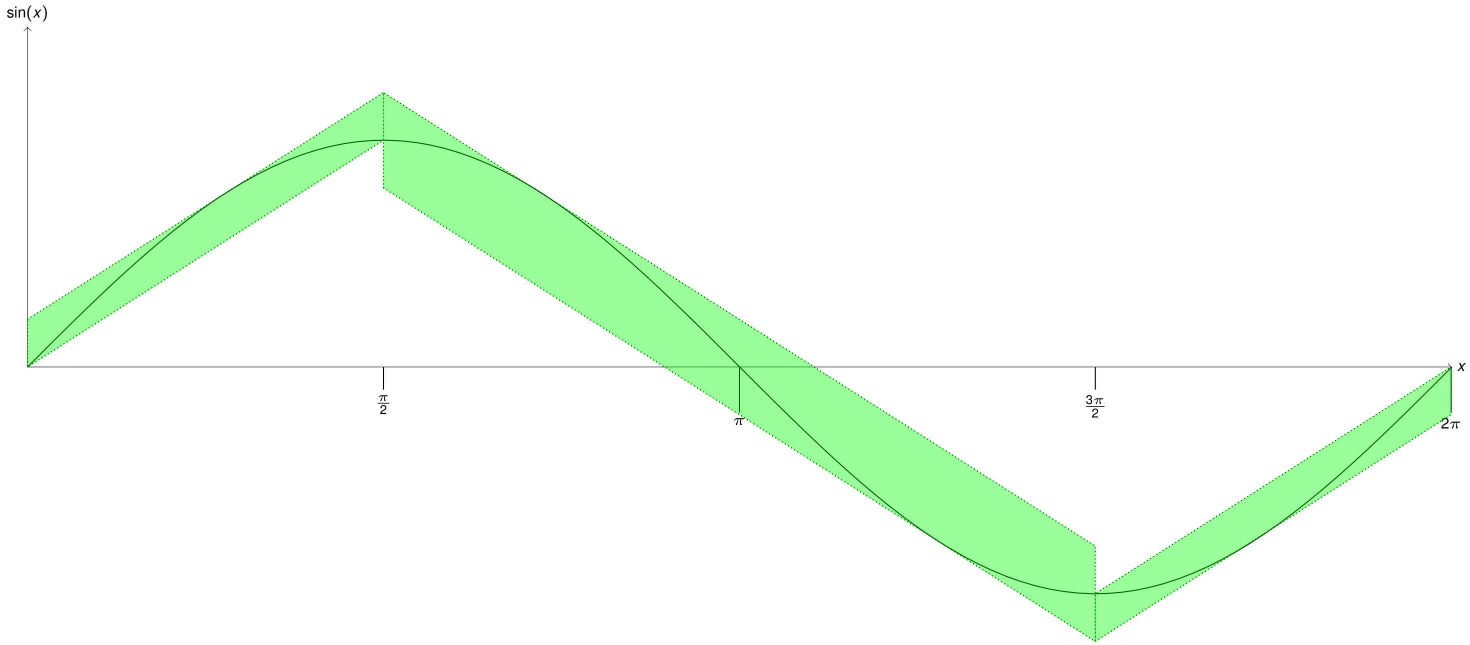
MIP-relaxations of MINLPs



MIP-relaxations of MINLPs



MIP-relaxations of MINLPs



Adaptive piecewise linear interpolation

Algorithm 1: Adaptive piecewise linear interpolation (Geißler, Morsi, Schewe 2013)

Data: A convex polytope $\mathcal{P} \subseteq \mathbb{R}^d$, a continuous function $f : \mathcal{P} \to \mathbb{R}$ and an upper bound $\epsilon > 0$ for the approximation error.

Result: A triangulation $\mathcal{S}$ of $\mathcal{P}$ corresponding to a piecewise lineare interpolation ϕ of f over $\mathcal{P}$ with $\phi(\mathbf{x}) = f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{V}(\mathcal{S})$ and $\phi = \phi_i$ for $S_i \in \mathcal{S}$ and $i = 1, \dots, n$.

Set $\mathcal{V} = \mathcal{V}(\mathcal{P})$;
Construct an initial triangulation $\mathcal{S}$ of $\mathcal{V}$ and the corresponding piecewise linear interpolation ϕ of f with $\phi(\mathbf{x}) = \phi_i(\mathbf{x})$ for $\mathbf{x} \in S_i$ for all $S_i \in \mathcal{S}$;
while $\exists S_i \in \mathcal{S}$, S_i *unmarked* **do**
    if $\epsilon(f, S) := \max_{\mathbf{x} \in S_i} |f(\mathbf{x}) - \phi_i(\mathbf{x})| > \epsilon$ **then**
        Add a point, where the maximal error is attained to $\mathcal{V}$;
        Set $\mathcal{S} \leftarrow \mathcal{S} \setminus \{S_i\}$;
        Update $\mathcal{S}$ according to the extended set of vertices $\mathcal{V}$;
    else
        Mark S_i;
    end
end
return $\mathcal{S}$

Computing the approximation error

- If f is convex or concave over S, this is easy!
- If f is indefinite over S computing $\epsilon(f, S)$ requires the solution of nonconvex NLPs to global optimality (in general NP-hard, cf. Murty & Kabadi 1985)

Computing the approximation error

- If f is convex or concave over S, this is easy!
- If f is indefinite over S computing $\epsilon(f, S)$ requires the solution of nonconvex NLPs to global optimality (in general NP-hard, cf. Murty & Kabadi 1985)

Definition

A function

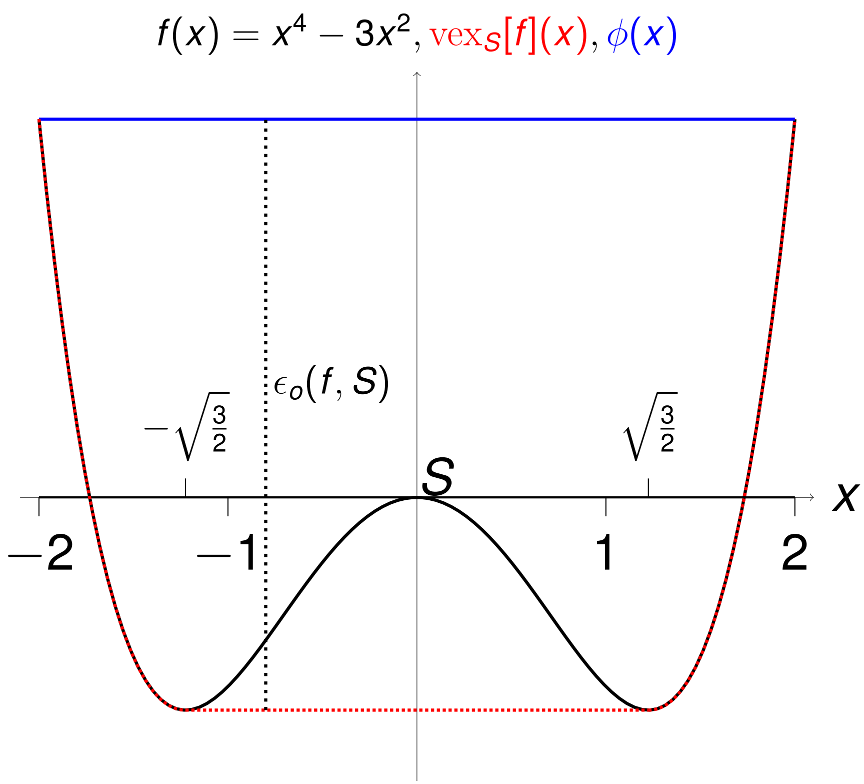
$$\mu \in \mathcal{U}(f, S) := \{\xi : S \to \mathbb{R} \, : \, \xi \text{ convex}, \, \xi(\mathbf{x}) \leq f(\mathbf{x}) \; \forall \mathbf{x} \in S\}$$

is called *convex underestimator* of f over S. The function $\mathrm{vex}_S[f] : S \to \mathbb{R}$ defined as

$$\mathrm{vex}_S[f](\mathbf{x}) := \sup\{\mu(\mathbf{x}) \; : \; \mu \in \mathcal{U}(f, S)\}$$

is called convex envelope of f over S.

Computing the approximation error



$$\mathcal{M}_0 = \left\{-\sqrt{\tfrac{3}{2}}, \sqrt{\tfrac{3}{2}}\right\}, \mathcal{N}_0 = \left[-\sqrt{\tfrac{3}{2}}, \sqrt{\tfrac{3}{2}}\right]$$

Computing the approximation error

Theorem (Geißler, Martin, Morsi, Schewe 2011)

Let $\mathcal{M}_o$ be the set of global maximizers for the overestimation of f by ϕ over S and let $\mathcal{N}_o$ be the set of points, where the global maximum of the overestimation of the convex envelope of f over S by ϕ is attained. Then,

$$\epsilon(f, S) = \epsilon(\mathrm{vex}_S[f], S) \text{ and } \mathcal{N}_o = \mathrm{conv}(\mathcal{M}_o).$$

Theorem (Geißler, Martin, Morsi, Schewe 2011)

Let ϕ be the linear interpolation of f over a d-simplex S. Then a point $\mathbf{x}^* \in \mathcal{M}_o$ can be obtained by solving at most d convex optimization problems in dimension $\leq d$.

Computing the convex envelopes

- For convex and concave functions computing the envelopes is trivial
- But in general this is NP-hard (Crama 1989)
- Convex envelopes are known for, e.g.,
  - univariate $\mathcal{C}^2$ functions over intervals (Maranas & Floudas 1995)
  - the bilinear function $f(x, y) = xy$ over boxes (McCormick 1976) and triangular domains (Locatelli & Schoen 2010)
- Combination of nonconvex functions (Ballerstein 2013)
- Our approximation algorithm works with any convex underestimator as, e.g., α-underestimators (Maranas & Floudas 1994)

- Observe, if convex envelopes are at hand, the triangulations are always refined, where the approximation errors attain their maxima

MIP-models for piecewise linear functions

- (linear) convex combination models
  - Linear number of cont. variables
  - Linear number of binaries
  - Linear number of constraints
- (logarithmic) convex combination models (Nemhauser, Vielma 2008)
  - Linear number of cont. variables
  - Logarithmic number of binaries
  - Logarithmic number of constraints
  - Locally ideal formulation
- incremental model (Markowitz, Manne 1957)
  - Linear number of cont. variables
  - Linear number of binaries
  - Linear number of constraints
  - Locally ideal formulation
- Also possible: SOS branching (Beale, Tomlin 1970)

The effect of branching: Incremental vs. Log

Figure: Branching induced by the leftmost and rightmost bit using the logarithmic convex combination model.

Figure: Branching induced by the 4th and 6th binary variable using the incremental model.

Incremental model – requirements

- Consider a d-variate plf on a polyhedral domain as a triangulation with d-simplices $\mathcal{S} = \{S_1, \ldots, S_n\}$ embedded in $\mathbb{R}^{d+1}$
- For an incremental model the following must hold:

(O1) The simplices in $\mathcal{S} = \{S_1, \ldots, S_n\}$ are ordered such that $S_i \cap S_{i+1} \neq \emptyset$ for $i = 1, \ldots, n-1$, and

(O2) for each simplex S_i its vertices $\overline{x}_0^{S_i}, \ldots, \overline{x}_d^{S_i}$ can be labeled such that $\overline{x}_d^{S_i} = \overline{x}_0^{S_{i+1}}$ holds for $i = 1, \ldots, n-1$.

Theorem (Geißler, Martin, Morsi, Schewe 2009)

In fixed dimension an ordering of the simplices and vertices that satisfies (O1) and (O2) can be computed in $\mathcal{O}(n^2)$ for any triangulation with n simplices.

Best known result so far by Wilson (1998) for $d = 2$ and domain homeomorphic to a disc.

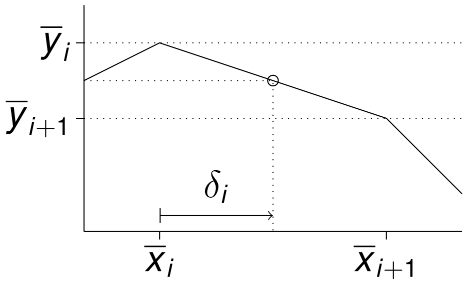
MIP-models for piecewise linear functions - Incremental method (Markowitz & Manne 1957)

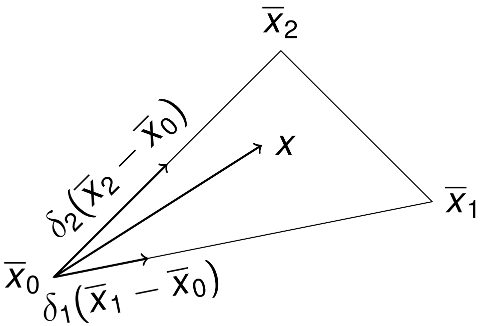
$$x = \overline{x}_0^{S_1} + \sum_{i=1}^{n} \sum_{j=1}^{d} \left(\overline{x}_j^{S_i} - \overline{x}_0^{S_i} \right) \delta_j^{S_i}, \qquad y = \overline{y}_0^{S_1} + \sum_{i=1}^{n} \sum_{j=1}^{d} \left(\overline{y}_j^{S_i} - \overline{y}_0^{S_i} \right) \delta_j^{S_i},$$

$$\sum_{j=1}^{d} \delta_j^{S_i} \le 1, \qquad \text{for } i = 1, \ldots, n,$$

$$\delta_j^{S_i} \ge 0, \qquad \text{for } i = 1, \ldots, n,\ j = 1, \ldots, d,$$

$$\sum_{j=1}^{d} \delta_j^{S_{i+1}} \le z_i, \quad z_i \le \delta_d^{S_i}, \quad z \in \{0,1\}^{n-1}.$$





From approximation to relaxation

$$x = \overline{x}_0^{S_1} + \sum_{i=1}^{n} \sum_{j=1}^{d} \left(\overline{x}_j^{S_i} - \overline{x}_0^{S_i} \right) \delta_j^{S_i}, \qquad y = \overline{y}_0^{S_1} + \sum_{i=1}^{n} \sum_{j=1}^{d} \left(\overline{y}_j^{S_i} - \overline{y}_0^{S_i} \right) \delta_j^{S_i} + e,$$

$$\sum_{j=1}^{d} \delta_j^{S_i} \leq 1, \qquad \text{for } i = 1, \ldots, n,$$

$$\delta_j^{S_i} \geq 0, \qquad \text{for } i = 1, \ldots, n, \ j = 1, \ldots, d,$$

$$\sum_{j=1}^{d} \delta_j^{S_{i+1}} \leq z_i, \quad z_i \leq \delta_d^{S_i}, \quad z \in \{0, 1\}^{n-1}.$$

$$\epsilon_u(f, S_1) + \sum_{i=1}^{n-1} z_i(\epsilon_u(f, S_{i+1}) - \epsilon_u(f, S_i)) \ \geq \ e,$$

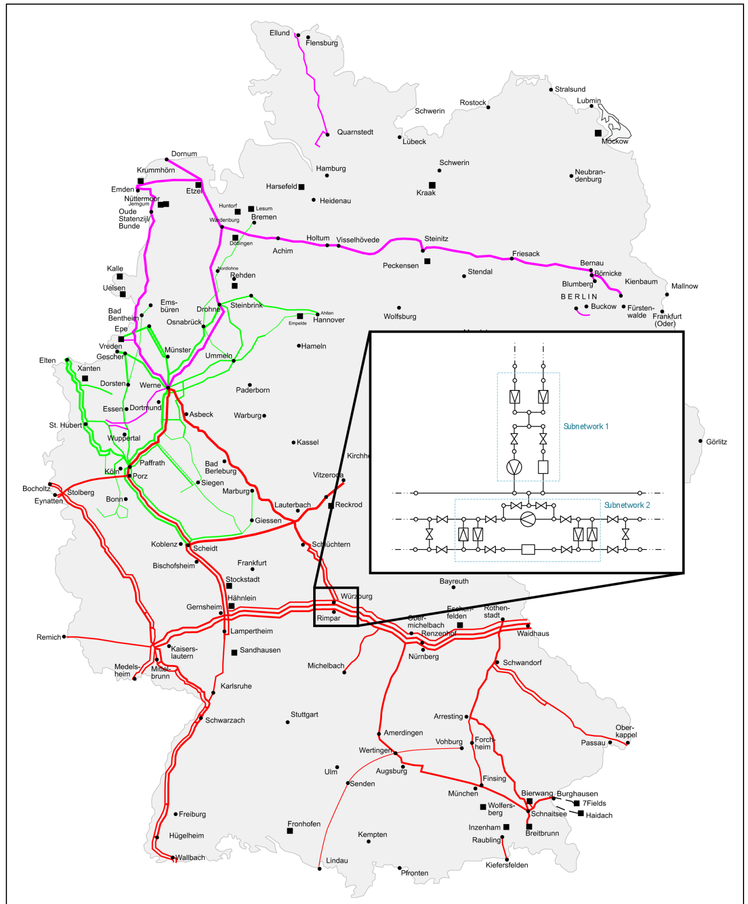
$$-\epsilon_o(f, S_1) - \sum_{i=1}^{n-1} z_i(\epsilon_o(f, S_{i+1}) - \epsilon_o(f, S_i)) \ \leq \ e.$$

Application: Gas Networks – Validation of Nominations

EDOM
Economics · Discrete Optimization · Mathematics

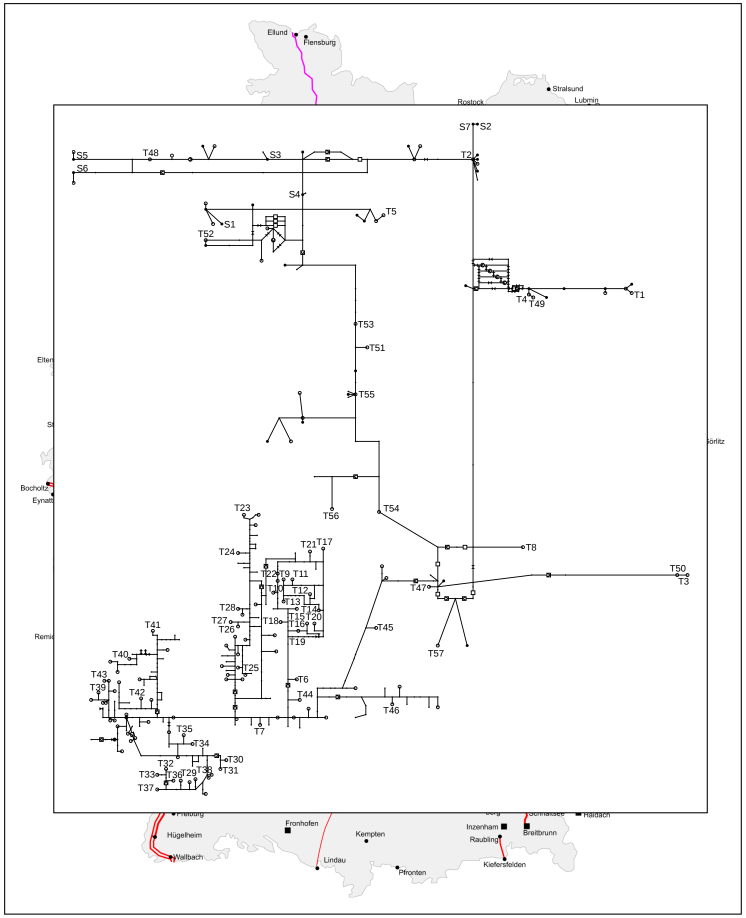
FRIEDRICH-ALEXANDER UNIVERSITÄT ERLANGEN-NÜRNBERG

Gas Networks

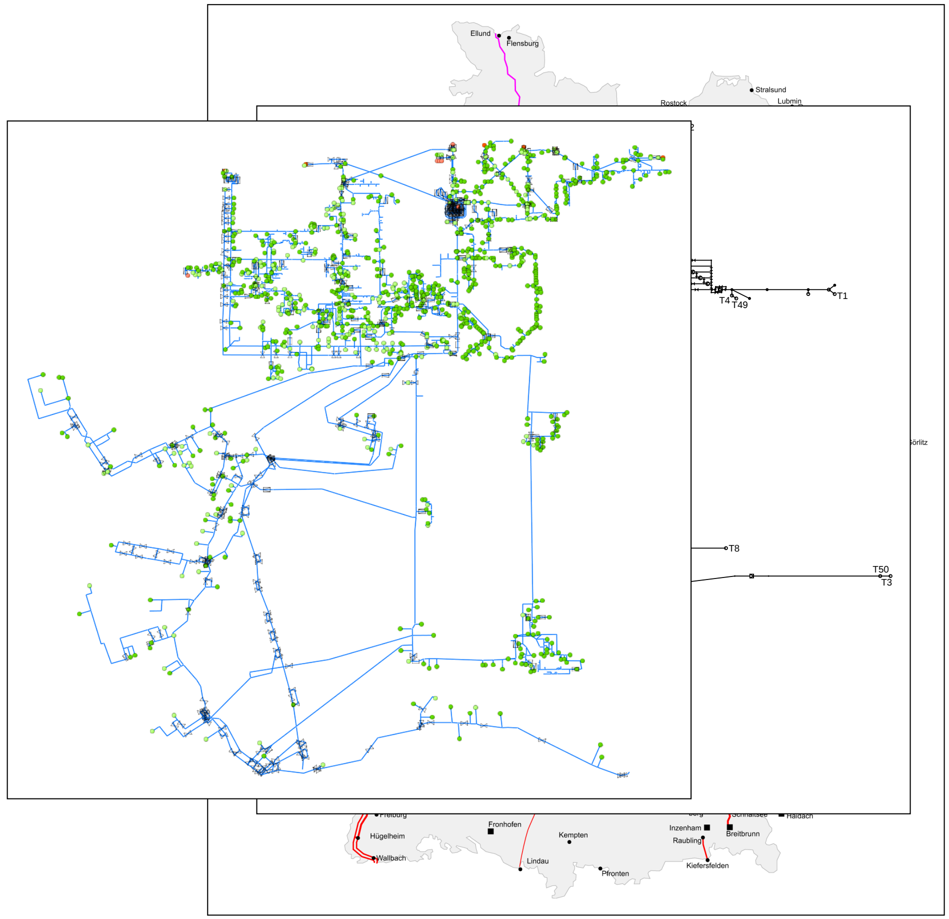


Gas Networks

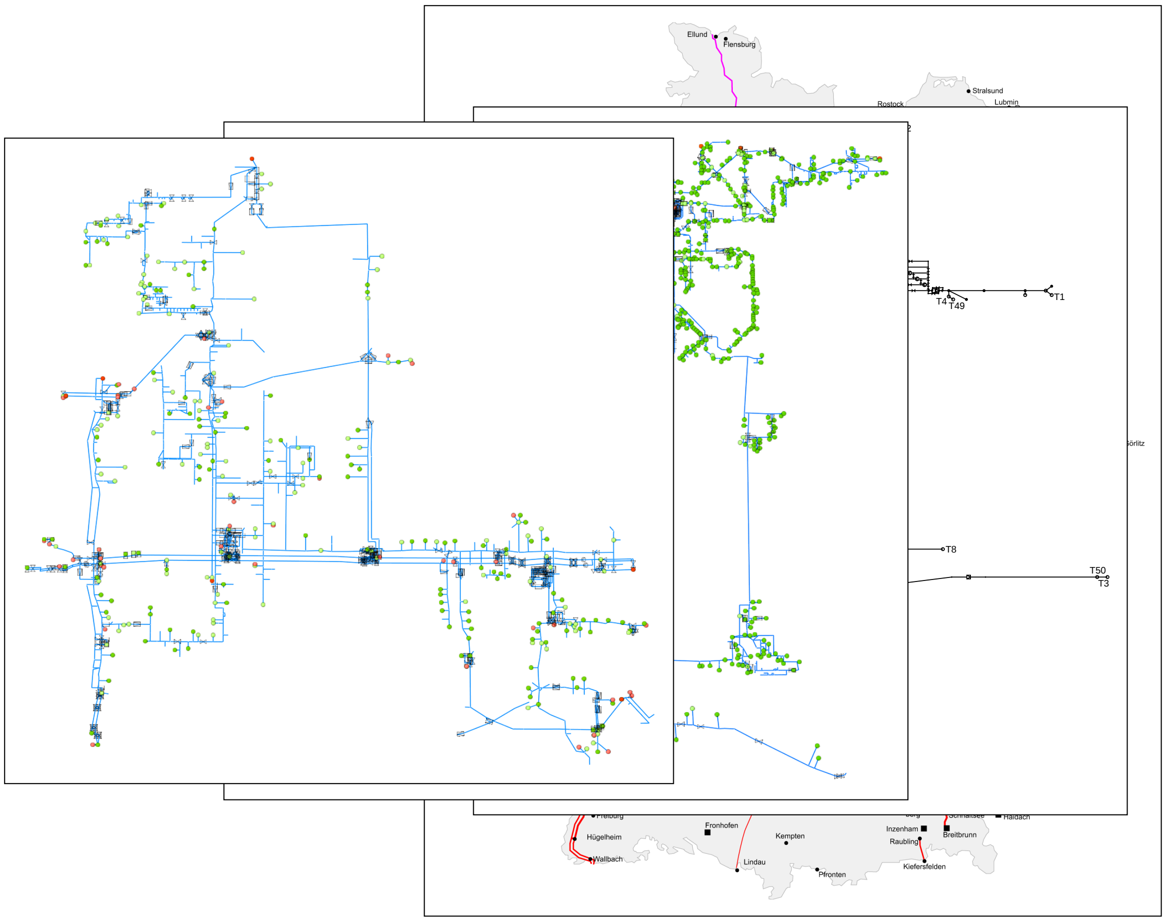
Gas Networks



Gas Networks



Gas Networks



Validation of nominations

- Given:
  - Specification of the network and the network elements
  - (balanced) set of inflows and outflows (=nomination)

Validation of nominations for gas networks

Exists a control of all active elements of a given gas network such that a given nomination could be realized (by the gas transport company) such that all technical, physical and legal constraints are satisfied?

- Constraints:

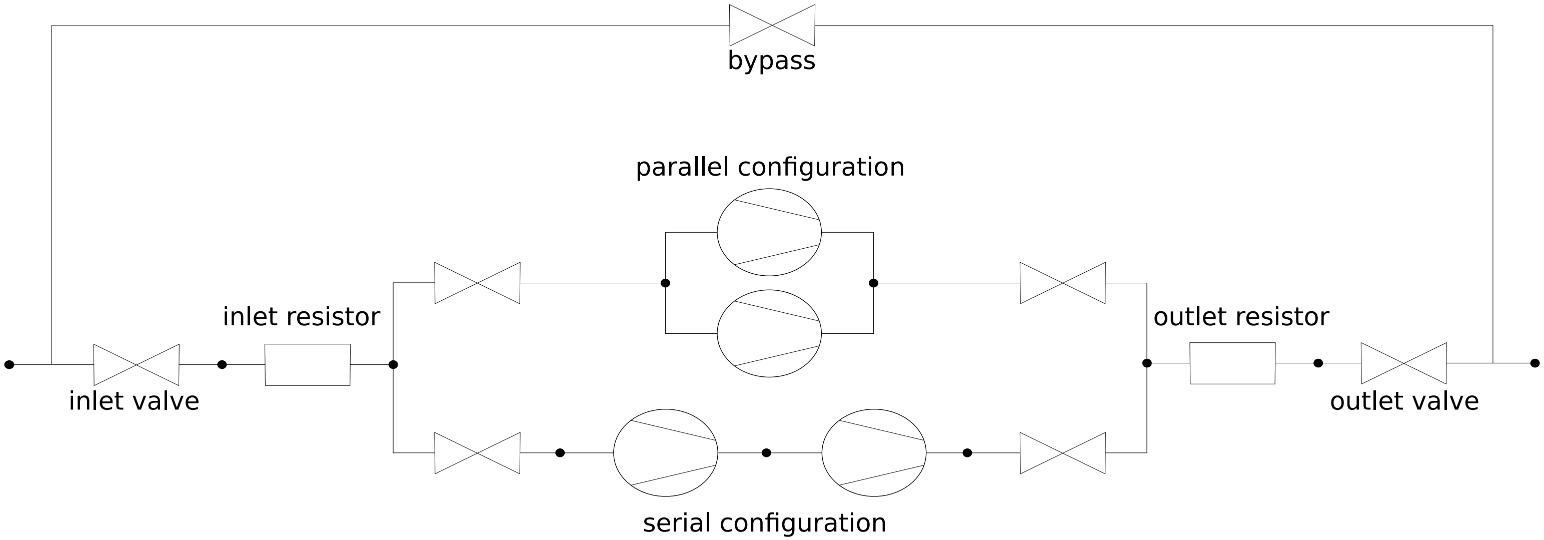
(technical) Min./max. pressures, flows, compressor powers, ...
(physical) (stationary) gas dynamics, flow conservation
(legal) Min./max. in-/outflows and pressures, ...

MINLP-model

- Variables:
  - Pressure p_i for all nodes $i \in \mathcal{V}$
  - Norm volume flow q_a for all arcs $a \in \mathcal{A}$
  - Compressor power P_a for all compressors $a \in \mathcal{A}$
- Nonlinear constraints:
  - Pressure loss over a pipe $a = (i, j)$: $p_j^2 = \left(p_i^2 - \Lambda_a |q_a| q_a \frac{e^{S_a}-1}{S_a}\right) e^{-S_a}$
  - Pressure loss over a resistor $a = (i, j)$: $p_i^2 - p_j^2 + |\Delta_{ij}|\Delta_{ij} = \frac{16 \rho_0 p_0 z_m}{\pi^2 z_0 T_0} \frac{\xi_a T}{D_a^4} |q_a| q_a, \; \Delta_{ij} = p_i - p_j$
  - Power consumption of a compressor unit $a = (i, j)$: $P_a = \frac{\kappa}{\kappa - 1} \frac{\rho_0 R T_i z_i}{\eta_{ad,a} m} \left(\left(\frac{p_j}{p_i} \right)^{\frac{\kappa-1}{\kappa}} - 1 \right) q_a$

MINLP-model

- Switching variables $s_a \in \{0,1\}$ for all active elements $a \in \mathcal{A}$
- Combinatorial constraints:
  - Valve: $s_a = 0 \rightarrow q_a = 0$, $s_a = 1 \rightarrow p_i = p_j$
  - Control valves: $s_a = 0 \rightarrow q_a = 0$, $s_a = 1 \rightarrow p_i - p_j \in [\Delta_a^-, \Delta_a^+]$
  - Compressors: $s_a = 0 \rightarrow q_a = 0$, $s_a = 1 \rightarrow p_j = f(P_a, q_a, p_i)$
  - Configurations: $\sum_{c \in C} s_c + s_{bypass} \leq 1$



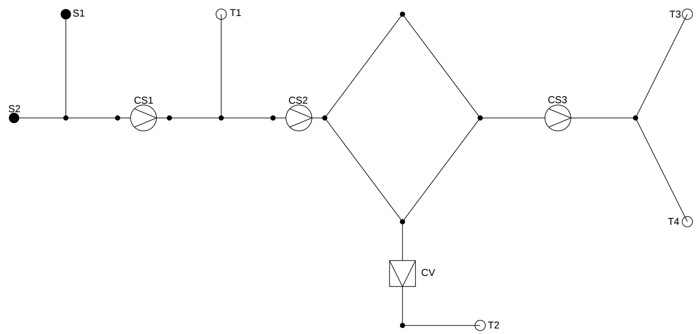
Numerical Results

EDOM
Economics · Discrete
Optimization · Mathematics

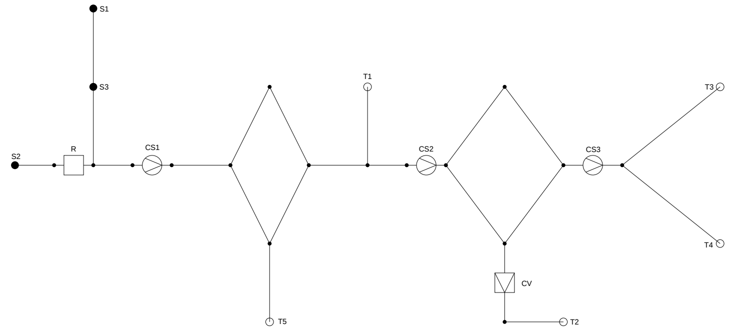
FRIEDRICH-ALEXANDER
UNIVERSITÄT
ERLANGEN-NÜRNBERG

Intances I - small test networks

- Net 1:

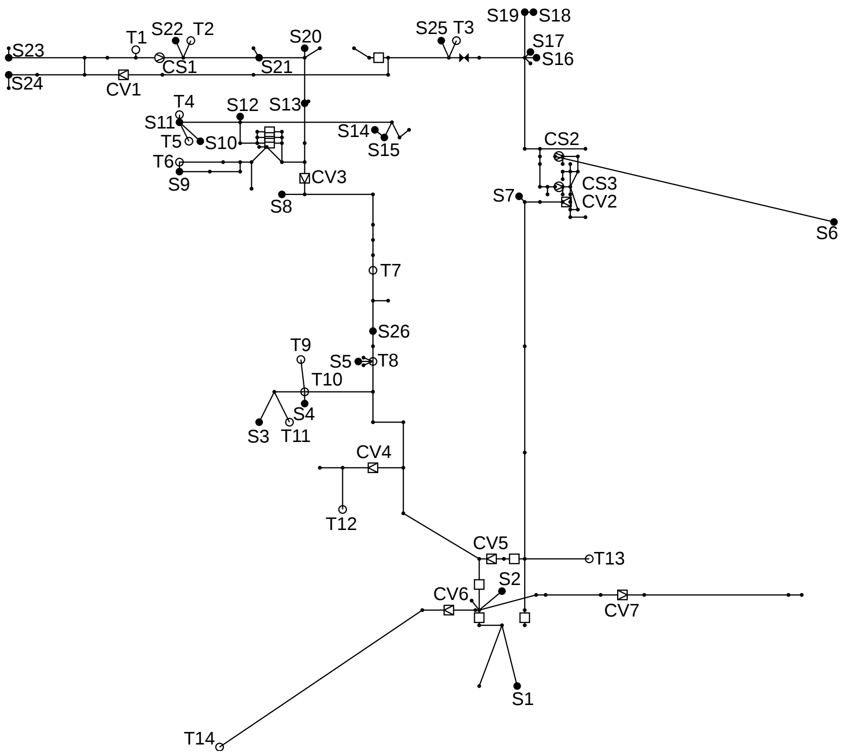


- Net 2:

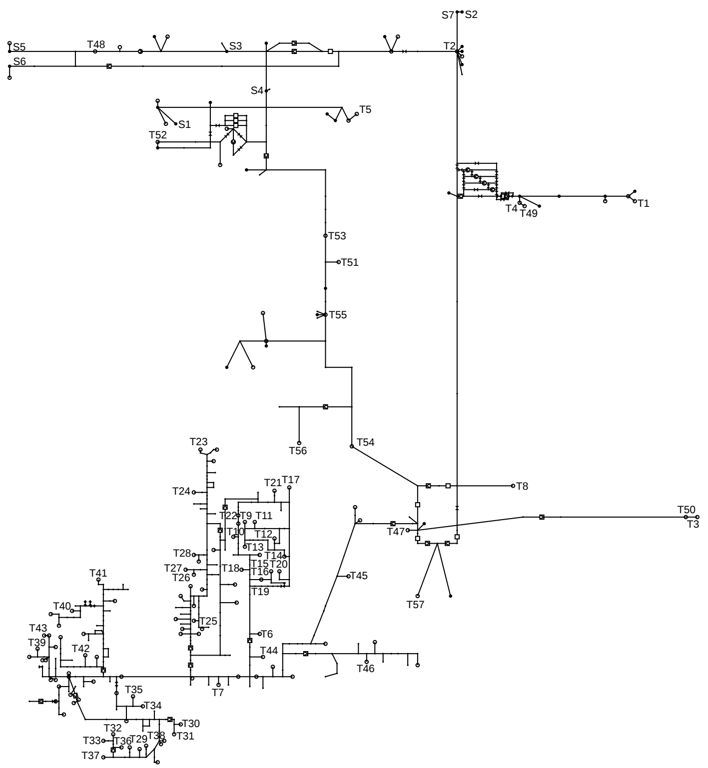


Instances II - real gas networks

- Net 3:



- Net 4:



Instances - overview

- Complexity of the test networks

net	$\lvert\mathcal{V}_S\rvert$	$\lvert\mathcal{V}_D\rvert$	$\lvert\mathcal{V}_I\rvert$	$\lvert\mathcal{A}_P\rvert$	$\lvert\mathcal{A}_S\rvert$	$\lvert\mathcal{A}_{CS}\rvert$	$\lvert\mathcal{A}_{CV}\rvert$	$\lvert\mathcal{A}_V\rvert$	$\lvert\mathcal{A}_R\rvert$
1	2	4	11	13	1	3	1	0	0
2	3	5	16	20	4	3	1	0	1
3	26	14	112	69	67	3	7	1	8
4	31	129	432	452	98	6	23	34	9

- One Validation of a nomination for each network
- Test system: 4 cores of a computer with two 6-core AMD Opteron 2435 processors with 2.6 GHz and 64GB RAM
- MIP-solver: Gurobi 4.0.1
- Time limit for MIP-solver: 10 min

Solving the validation of nominations problem

- Approach
  1. Solve the MIP-relaxation
  2. Fix the discrete decision variables in the MINLP-model according to the MIP-solution
  3. Solve the remaining NLP-model
- If the NLP is feasible $\rightarrow$ Solution to the validation of nominations problem
- (MI)NLP-solvers: Baron 11.1.0, SCIP 3.0
- Max. error in pressure loss equations: 1.0/2.5/5.0/10.0 bar

Validation of nominations - results

net	ϵ	cont	bin	cons	t_{MIP}	feas	t_{NLP}
1	10.0	377	42	685	0.01s	y	0.11s
1	5.0	380	45	694	0.01s	y	0.15s
1	2.5	387	52	714	0.01s	y	0.06s
1	1.0	423	88	823	0.02s	y	0.20s
2	10.0	450	67	859	0.05s	y	0.26s
2	5.0	479	96	946	0.09s	y	0.20s
2	2.5	543	160	1138	0.11s	y	0.09s
2	1.0	816	433	1957	0.13s	y	0.26s
3	10.0	2099	418	3868	1.24s	n	12.94s
3	5.0	2412	713	4807	1.51s	y	1.48s
3	2.5	3058	1377	6745	6.03s	y	1.32s
3	1.0	5185	3504	13126	22.04s	y	1.59s
4	10.0	4825	1663	10994	21.65s	n	41.33s
4	5.0	6012	2850	14555	51.26s	y	30.83s
4	2.5	8433	5217	21818	132.96s	y	36.65s
4	1.0	16343	13181	45548	600.00s	-	-

Validation of nominations - results

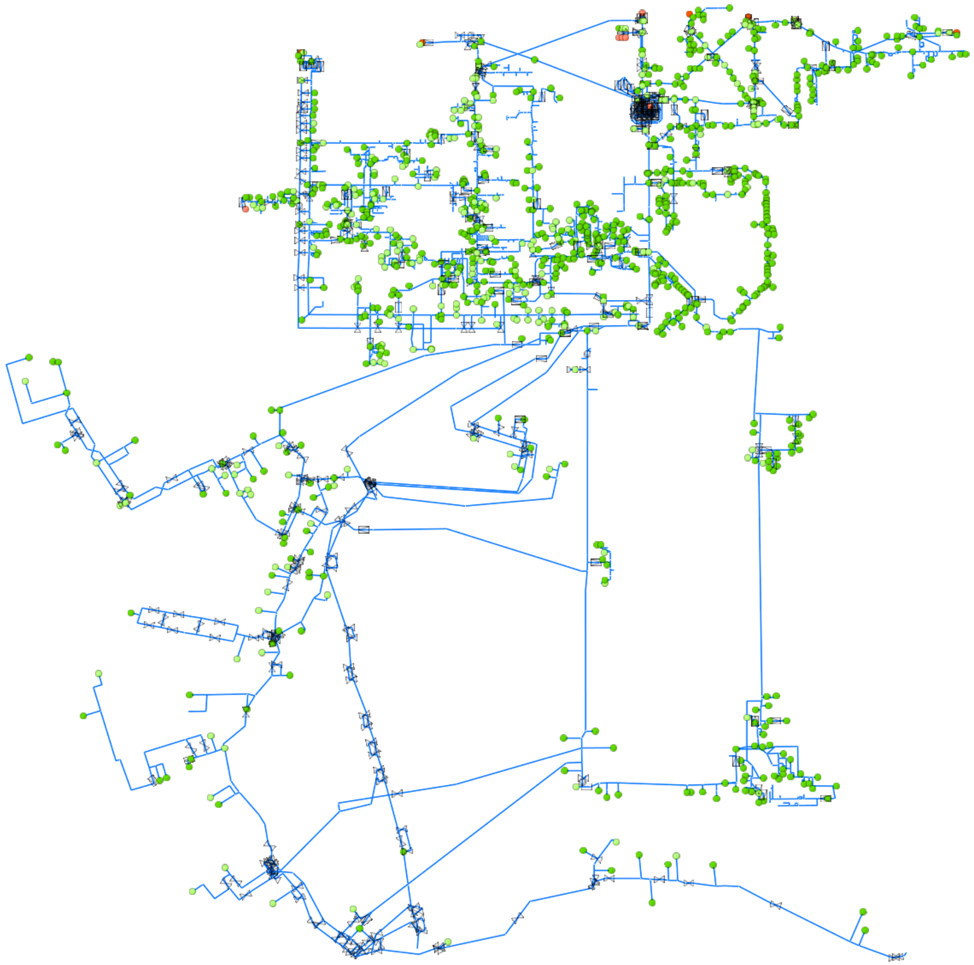
net	ϵ	cont	bin	cons	t_{MIP}	feas	t_{NLP}
1	10.0	377	42	685	0.01s	y	0.11s
1	5.0	380	45	694	0.01s	y	0.15s
1	2.5	387	52	714	0.01s	y	0.06s
1	1.0	423	88	823	0.02s	y	0.20s
2	10.0	450	67	859	0.05s	y	0.26s
2	5.0	479	96	946	0.09s	y	0.20s
2	2.5	543	160	1138	0.11s	y	0.09s
2	1.0	816	433	1957	0.13s	y	0.26s
3	10.0	2099	418	3868	1.24s	n	12.94s
3	5.0	2412	713	4807	1.51s	y	1.48s
3	2.5	3058	1377	6745	6.03s	y	1.32s
3	1.0	5185	3504	13126	22.04s	y	1.59s
4	10.0	4825	1663	10994	21.65s	n	41.33s
4	5.0	6012	2850	14555	51.26s	y	30.83s
4	2.5	8433	5217	21818	132.96s	y	36.65s
4	1.0	16343	13181	45548	600.00s	-	-

Solving validation of nominations MINLPs

net	Baron (MINLP)	SCIP (MINLP)	MIP	NLP	MIP+NLP
1	$<$1s	$<$1s	$<$1s	$<$1s	$<$1s
2	$<$1s	$<$1s	$<$1s	$<$1s	$<$1s
3	456s	2s	2s	1s	3s
4	$>$1h	$>$1h	51.26s	30.83s	82.09s

Network 5 (Open Grid Europe, L-Gas Germany)

- 13 entries
- 1.062 exits
- 3.632 pipes
- 26 resistors
- 305 valves
- 120 control valve stations
- 12 compressor stations
- 25.000 variables (5.000 binary)



Computing time on 51 *expert scenarios*:

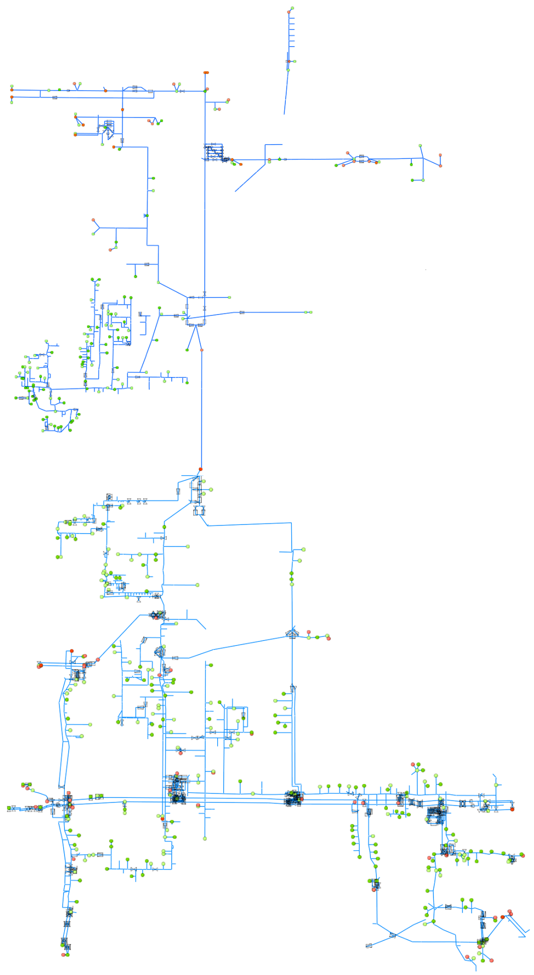
- 11 min to 13.5 hours
- average: 2.5 hours

Network 6 (Open Grid Europe, H-Gas Germany)

- 78 entries
- 395 exits
- 1.588 pipes
- 56 resistors
- 264 valves
- 101 control valve stations
- 35 compressor stations
- 35.000 variables
  (14.000 binary)

Computing time on 29 *expert scenarios*:

- 3 to 46 hours
- average: 17 hours



Summary

- Purely polyhedral view on (mixed-integer) nonlinear problems
- Validation of nominations problem can be solved much faster than with state-of-the-art MINLP-solvers
- Convincing computation results, even for large-scale real-life instances
- State-of-the-art MIP-solvers may be used to solve MINLPs, if the number of variables of each nonlinear function is small

The Future

- The instationary case
- New Reserach Grant: Cooperate Reseach Center CRC 154 supported by the German Science Foundation (DFG)

Thank you for coming