Mixed-integer methods for polynomial optimization and application to energy networks

Carl Eggen

University of Konstanz

27th March 2023

Outline

Motivation

How can climate targets be achieved in urban energy supply?

Contribution of mathematics:

Algorithms for calculating optimal energy networks

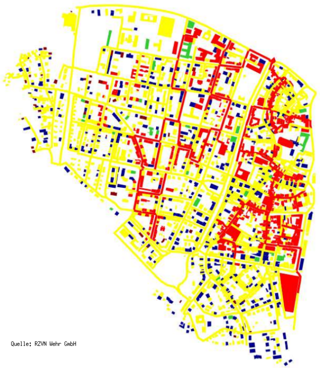


Figure 1: Example of an energy network for Konstanz

Network model [Lu21]

Minimize the total costs of energy supply considering the following aspects:

- energy sources, i.e. natural gas and electricity (soon: district heating)
- energy demand
- network expansion
- power balance
- CO_2-limit
- Renovation progress
- Nonlinear physical laws for the individual energy carriers (e.g. Ohm's law for electricity)

Optimization of the network model

The network model yields a mixed-integer nonlinear optimization problem, in short MINLP

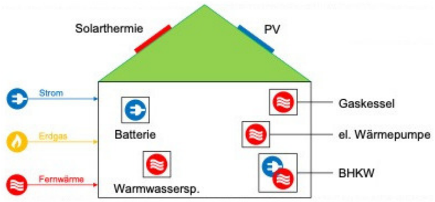


Figure 2: How should houses be supplied with energy?

- integer variables: gas pipe or district heating?
- continuous variables: How much electricity/gas/district heating needs to be supplied?

Optimization problem

The network model yields mixed-binary polynomial optimization problems (MBPOP) of the form

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \ & f(x) \\ \text{s.t. } & g_j(x) \geq 0 \text{ for } j \in [m], \\ & x_i \in \{0,1\} \text{ for } i \in I, \end{aligned} \tag{MBPOP}$$

where

- $f, g_1, \ldots, g_m \in \mathbb{R}[x]$ are polynomials in $x_1, \ldots, x_n$ variables,
- $x = (x_1, \ldots, x_n)$ is the vector of variables,
- $I \subseteq [n]$ index set of binary variables in x,
- $S(g) := \{x \in \mathbb{R}^n \mid g_j(x) \geq 0\}$ is box constrained,
- $S(g,h) := \{x \in \mathbb{R}^n \mid g_j(x) \geq 0, h_i(x) = x_i(x_i - 1) = 0\}$.

For solving MBPOPs different relaxations techniques can be used.

Moment hierarchy: Reformulation [Las01; Lau09]

$$f^{\min} := \inf_{x \in S(g)} f(x) \tag{POP}$$

Rewrite (POP)

$$\inf_{x \in S(g)} f(x) = \inf_{\mu \in \mathcal{P}_{S(g)}} \int_{S(g)} f(x)\, \mu(dx)$$
$$= \inf_{\mu \in \mathcal{P}_{S(g)}} \sum_{\alpha} f_\alpha \int_{S(g)} x^\alpha\, \mu(dx) = \inf_{y \in \mathcal{M}_{S(g)}} \mathrm{vec}(f)^\top y,$$

where

$$\mathcal{P}_{S(g)} = \{\mu \text{ probability measure on } \mathbb{R}^n \mid \mathrm{supp}(\mu) \subseteq S(g)\},$$

and

$$\mathcal{M}_{S(g)} = \{y \in \mathbb{R}^{\mathbb{N}^n} \mid \exists \mu \in \mathcal{P}_{S(g)} : y_\alpha = \int_{S(g)} x^\alpha\, \mu(dx)\}.$$

Moment hierarchy: Relaxation

Lemma

If $y \in \mathbb{R}^{\mathbb{N}^n}$ is the sequence of moments of a measure μ, then $M(y) \succeq 0$. Let $g \in \mathbb{R}[x]$. If $\operatorname{supp}(\mu) \subseteq \{x \in \mathbb{R}^n \mid g(x) \geq 0\}$, then $M(gy) \succeq 0$.

Relax the constraint $y \in \mathcal{M}_{S(g)}$ by the weaker constraint of positive semidefitness of the (infinite) moment matrix $M(y)$ and localizing matrices $M(g_j y)$

$$
\begin{aligned}
f^{\text{MOM}} = \inf_{y \in \mathbb{R}^{\mathbb{N}^n}} \ & \operatorname{vec}(f)^\top y \\
\text{s.t. } & y_0 = 1, M(y) \succeq 0 \\
& M(g_j y) \succeq 0 \text{ for } j = 1, \dots, m
\end{aligned}
$$

$$f^{\text{MOM}} \leq f^{\min}$$

Moment hierarchy: Truncation

Truncating the moment matrices yields a hierarchy of SDPs

$$f_d^{\text{MOM}} = \inf_{y \in \mathbb{R}^{\mathbb{N}^n_{2d}}} \ \text{vec}(f)^\top y \qquad (\text{MOM}_d)$$

$$\text{s.t. } y_0 = 1, M_d(y) \succeq 0$$

$$M_{d-d_j}(g_j y) \succeq 0 \text{ for } j = 1, \ldots, m$$

for any $d \geq \lceil \frac{\deg(f)}{2} \rceil$ with $d_1 := \lceil \frac{\deg(g_1)}{2} \rceil, \ldots, d_m := \lceil \frac{\deg(g_m)}{2} \rceil$.

$$f_d^{\text{MOM}} \leq f_{d+1}^{\text{MOM}} \leq f^{\text{MOM}} \leq f^{\min}$$

Further properties

- **Duality:**
  The Lagrangian dual SOS_d of MOM_d uses sums of squares representations, $f_d^{\text{SOS}} \leq f_d^{\text{MOM}}$ and no duality gap, if $\text{int}(S(g)) \neq \emptyset$.
- **Convergence:**
  $\lim_{d\to\infty} f_d^{\text{SOS}} = f^{\min}$ via Putinar's Positivstellensatz from real algebraic geometry.
- **Optimality certificate:**
  If $\text{rank}\, M_d(y^d) = \text{rank}\, M_{d+1}(y^{d+1})$, then $f_d^{\text{MOM}} = f^{\min}$ holds (y^d solution of MOM_d).
- **Extracting global minimizers:**
  If rank condition is satisfied, compute finite atomic meassure for truncated moment sequence y^d.
- **Unique minimizer:**
  x^* unique solution of MBPOP, then $\lim_{d\to\infty} y_{e_i}^d = x_i^*$

Some practical problems of the moment hierarchy

- Size of matrices is proportional to $\binom{n+d}{d}$ in MOM_d (n number of variables).
- Sparsity can be exploited. But if n is big, even MOM_2 can be too large.
- In most practical application: no global optimality and no extraction of global minimizers possible.
- Even no feasible point can be extracted from solutions of MOM_d.
- **BUT**: First moments can be used as initial point for an local NLP-Solver.

Feasiblity pump: Finding feasible points

Motivation

Feasible points yield upper bounds for optimal value
$\rightarrow$ essential for a Branch-and-Bound framework

Feasibility pump: Heuristic for finding feasible points

Feasiblity pump: Basic idea [FGL05; BLS19]

Generating two sequences of points which (hopefully) converge to a feasible point, i.e.

$$x^{k,R} \to x^{\text{fea}} \leftarrow x^{k,I}$$

for $k \to \infty$ with

- $(x^{k,R})_k$ sequence of solutions of a relaxation of MBPOP with relaxed feasible set $S(g,h) \subseteq R$.
- $(x^{k,I})_k$ sequence of points fulfilling the binary condition, i.e. $x_i^{k,I} \in \{0,1\}$ for $i \in I$,

Feasiblity pump: Basic idea [FGL05; BLS19]

Generating two sequences of points which (hopefully) converge to a feasible point, i.e.

$$x^{k,R} \to x^{\mathrm{fea}} \leftarrow x^{k,I}$$

for $k \to \infty$.
Starting from a solution of a relaxation $x^{0,R}$ new points are generated by alternately

- rounding a feasible point of the relaxation and
- projecting the rounded, i.e. integral, solution onto the closest point in the relaxed feasible set R.

Feasiblity pump: Three main ingredients for $R = S(g)$

There are three main ingredients for the feasibility pump:

- **Rounding:** Rounding of $x^{k,S(g)}$ to nearest integral point $x^{k,I}$.
- **Projecting:** Solving

$$\min_{x \in S(g)} \Delta_\rho(x, x^{k,I}) := (1-\rho) \sum_{i \in I} |x_i - x_i^{k,I}| + \rho c f(x)$$

  yields $x^{k,S(g)}$ with $\rho \in [0,1]$ and $c \in \mathbb{R}_+$.
- **Perturbing:** Cycling might occur, i.e. $x^{k-l,I} = x^{k,I}$ for some $l \in \mathbb{N}$. Then perturb some of the binary entries of $x^{k,I}$.

Feasibility Pump: Basic idea

Algorithm 1 Ideal feasibility pump for MBPOP

Require: MBPOP $\equiv \min \{f(x) \mid x \in S(g),\ \forall i \in I\colon x_i \text{ binary}\}$

Ensure: a feasible solution x^{fea} (if found)

```
x^{0,S(g)} = arg min{f(x) | x ∈ S(g)}
while not termination condition do
    if x^{k,S(g)} is integer then
        return x^fea = x^{k,S(g)}
    else
        x^{k,I} = round(x^{k,S(g)})
    end if
    if cycle detected then
        perturb(x^{k,I})
    end if
    x^{k,S(g)} = arg min_{x∈S(g)} Δ_ρ(x, x^{k,I})
end while
```

$$x^{0,S(g)} = \arg\min\{f(x) \mid x \in S(g)\}$$

$$x^{k,I} = \texttt{round}(x^{k,S(g)})$$

$$x^{k,S(g)} = \arg\min_{x \in S(g)} \Delta_\rho(x, x^{k,I})$$

Feasibility pump: Basic idea

Algorithm 1 Ideal feasibility pump for MBPOP

Require: MBPOP $\equiv \min \{f(x) \mid x \in S(g),\ \forall i \in I\colon x_i \text{ binary}\}$
Ensure: a feasible solution x^{fea} (if found)

$x^{0,S(g)} = \arg\min\{f(x) \mid x \in S(g)\}$
while *not termination condition* **do**
  if $x^{k,S(g)}$ is integer **then**
    return $x^{\text{fea}} = x^{k,S(g)}$
  else
    $x^{k,I} = \texttt{round}(x^{k,S(g)})$
  end if
  if cycle detected **then**
    $\texttt{perturb}(x^{k,I})$
  end if
  $x^{k,S(g)} = \arg\min_{x \in S(g)} \Delta_\rho(x, x^{k,I})$
end while

Feasiblity Pump: Solving the NLPs

How to solve the nonconvex NLPs? Two Options:

- using a local NLP-Solver (e.g. `Ipopt`)
  BUT: only local solution, but feasible for the NLP.
- using the moment/SOS hierarchies for global optimization
  BUT: (extracted) solution of a moment relaxation is *neither* integer *nor* feasible for the NLP, but can be used as an inital point for a local NLP-Solver.

Feasiblity pump: Adaption to MBPOPs

- Moment relaxation for finding an initial feasible candidate.
- After rounding: Fix the integer variables and enforce nonlinear constraints by a local NLP-Solver.
- If local NLP-solver fails, perturb binary entries.
- Projecting: Solve moment relaxation with objective

$$\Delta_\rho(y, x^{k,l}) = (1-\rho)\sum_{i\in I} |y_{e_i} - x_i^{k,l}| + \rho c\, \mathrm{vec}(f)^\top y,$$

where y_{e_i} is the variable associated to the first moment $\int x_i d\mu$.

Feasiblity pump: Adaption to MBPOPs

Algorithm 2 Feasibility pump for MBPOP

Require: $\text{MBPOP} \equiv \min \{f(x) \mid x \in S(g),\ \forall i \in I\colon x_i \text{ binary}\},\ d$

Ensure: a feasible solution x^{fea} (if found)

get solution $y^{0,S(g,h)}$ of MOM_d

while *not termination condition* **do**

  extract solution candidate $x^{k,S(g,h)}$ from $y^{k,S(g,h)}$

  $x^{k,I} = \texttt{round}(x^{k,S(g,h)})$

  $x^{\text{fea}}, \text{stat} = \texttt{solve\_local}(\min_{x \in S(g)} \{f(x) \mid \forall i \in I : x_i = x_i^{k,I}\})$

  if stat is `LOCALLY FEASIBLE` **then**

    return x^{fea}

  else

    $\texttt{perturb}(x^{k,I})$

  end if

  get solution $y^{k,S(g,h)}$ of MOM_d with objective $\Delta_\rho(y, x^{k,I})$

end while

Numerical results

Set Up: minimal order of moment relaxation, sparsity was exploited.

- On 39 of 48 problems from the MINLPLib the FP was successful in under 100 iterations.
- Compared to `SCIP` the FP yields better and faster feasible points for some instances.
- Random perturbation makes comparison difficult!

Numerical Results: Challenging Instances from MINLPLib

Name	#variables			#constraints		
	cont.	bin.	total	linear	nonlin.	total
chp_shorttermplan1a	864	144	1008	1684	384	2068
genpooling_meyer04	63	55	118	126	15	141
ringpack_10_1	20	50	70	55	330	385
ringpack_20_1	40	175	215	210	2337	2547
ringpack_20_3	40	213	253	173	3055	3228
ringpack_30_1	60	373	433	465	7433	7898
sfacloc1_3_80	231	62	293	2146	15	2161
sfacloc1_3_90	231	30	261	406	15	421
sfacloc1_3_95	226	9	233	266	15	281
sfacloc1_4_90	293	30	323	479	15	494
sfacloc1_4_95	286	9	295	339	15	354

Table 1: Examples from `MINLPLib`

Numerical Results: Comparison to SCIP

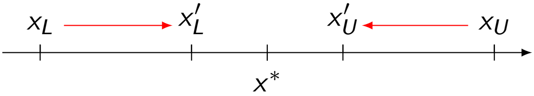
Name	FP		SCIP		
	t(s)	value	t(s)	value	#sol
chp_shorttermplan1a	52.9	331.1	**4.3**	**313.3**	1
genpooling_meyer04	**2.3**	**2.126e6**	3.5	2.279e6	1
ringpack_10_1	3.1	**-6.4**	**0.2**	-2.1	4
ringpack_20_1	**10.7**	**-30.1**	17.3	-8.9	5
ringpack_20_3	**14.4**	**-10.4**	57.2	-8.3	14
ringpack_30_1	**5.9**	**-6.3**	11.9	-2.1	4
sfacloc1_3_80	3.4	10.25	**3.0**	**9.2**	1
sfacloc1_3_90	2.7	12.3	**1.2**	**11.8**	1
sfacloc1_3_95	**0.9**	**12.7**	1.0	20.0	1
sfacloc1_4_90	3.7	**11.4**	**1.6**	16.3	1
sfacloc1_4_95	**1.6**	**11.8**	1.9	16.7	1

Table 2: Comparison to SCIP

Optimality-based bound tightening

What can we do with feasible points besides using them for upper bounds in Branch-and-Bound Algorithms?

Tightening bounds to shrink the feasible set



BENEFIT: Tighter bounds $\Longrightarrow$ lower order ?

Optimality-based bound tightening [Bel+13]

Let x^{fea} be a feasible point. Then solving

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \ & -x_j \\ \text{s.t. } & f(x) \le f(x^{\mathrm{fea}}), \\ & g_j(x) \ge 0 \text{ for } j \in [m], \\ & x_i \in \{0,1\} \text{ for } i \in I, \end{aligned}$$

for $j \in [n] \setminus I$ yields tighter **lower** bound for x_j.

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \ & x_j \\ \text{s.t. } & f(x) \le f(x^{\mathrm{fea}}), \\ & g_j(x) \ge 0 \text{ for } j \in [m], \\ & x_i \in \{0,1\} \text{ for } i \in I, \end{aligned}$$

for $j \in [n] \setminus I$ yields tighter **upper** bound for x_j.

Optimality-based bound tightening: A simple example

ex3_1_1 from MINLPLib:

$$
\begin{aligned}
\min_{x\in\mathbb{R}^8}\ & x_1 + x_2 + x_3 \\
\text{s.t. } & 0.0025(x_4 + x_6) \leq 1, \\
& 0.0025(x_5 + x_7 - x_4) \leq 1, \\
& 0.01(x8 - x5) \leq 1, \\
& 100x_1 - x_1x_6 + 833.33252x_4 \leq 83333.333, \\
& x_2x_4 - x_2x_7 - 1250x_4 + 1250x_5 \leq 0, \\
& x_3x_5 - x_3x_8 - 2500x_5 \leq -1250000, \\
& 100 \leq x_1 \leq 10000, 1000 \leq x_2, x_3 \leq 10000, \\
& 10 \leq x_4, x_5, x_6, x_7, x_8 \leq 1000.
\end{aligned}
$$

Optimality-based bound tightening: Numerical results

New bounds for ex3_1_1 with OBBT:

Variable	sol	orig. bounds		OBBT ($d = 1.5$)	
	x^*	x_L	x_U	x'_L	x'_U
x_1	579.19	100	10000	100	**2588.0**
x_2	1360.13	1000	10000	1000	**3488.0**
x_3	5109.92	1000	10000	**3470.0**	**5949.2**
x_4	182.01	10	1000	10	**251.3**
x_5	295.6	10	1000	**262.0**	**360.1**
x_6	217.99	10	1000	10	**390.0**
x_7	286.4	10	1000	**100.2**	**389.4**
x_8	395.6	10	1000	**362.0**	**461.2**
value	7049.25	2100.0 (0.01s)		**4569.8** (8.4s)	

Table 3: OBBT for ex3_1_1 with optimal value, value of moment relaxation of minimal order ($d = 1$) and time

Optimality-based bound tightening: Numerical results

New bounds for ex3_1_1 with OBBT:

Variable	sol	orig. bounds		OBBT ($d = 2$)	
	x^*	x_L	x_U	x'_L	x'_U
x_1	579.19	100	10000	**479.0**	**691.7**
x_2	1360.13	1000	10000	**1217.6**	**1503.5**
x_3	5109.92	1000	10000	**4967.6**	**5253.5**
x_4	182.01	10	1000	**173.0**	**190.7**
x_5	295.6	10	1000	**289.9**	**301.3**
x_6	217.99	10	1000	**209.3**	**227.0**
x_7	286.4	10	1000	**277.8**	**294.5**
x_8	395.6	10	1000	**389.9**	**401.3**
value	7049.25	2100.0 (0.01s)		**7035.5** (35.2s)	

Table 4: OBBT for ex3_1_1 with optimal value, value of moment relaxation of order $d = 2$ and time

Optimality-based bound tightening: Drawbacks

- `ex3_1_1` can be solved without OBBT to global optimality by the moment relaxation of order $d = 3$ in **0.16s**!
- OBBT is very costly (2n SDPs to solve)
- OBBT with order $d = 1.5$ is still too high for challenging problems from the MINLPLib, does not improve lower bound significantly

$\longrightarrow$ applying advanced strategies to save OBBT ?

Outlook

- Further improvement of feasibility pump and OBBT
- Development of a Branch-and-Bound algorithm with the moment/sos hierarchy as main relaxation method
- Integration of moment/sos hierarchy into existing Branch-and-Bound frameworks

Thank you!

References

[Bel+13] Pietro Belotti, Christian Kirches, Sven Leyffer, Jeff Linderoth, James Luedtke, and Ashutosh Mahajan. “Mixed-integer nonlinear optimization”. In: *Acta Numer.* 22 (2013), pp. 1–131.

[BLS19] Timo Berthold, Andrea Lodi, and Domenico Salvagnin. “Ten years of feasibility pump, and counting”. In: *EURO J. Comput. Optim.* 7.1 (2019), pp. 1–14.

[FGL05] Matteo Fischetti, Fred Glover, and Andrea Lodi. “The feasibility pump”. In: *Math. Program.* 104.1, Ser. A (2005), pp. 91–104.

[Las01] Jean B. Lasserre. “Global optimization with polynomials and the problem of moments”. In: *SIAM J. Optim.* 11.3 (2000/01), pp. 796–817.

[Lau09] Monique Laurent. “Sums of squares, moment matrices and optimization over polynomials”. In: *Emerging applications of algebraic geometry.* Vol. 149. IMA Vol. Math. Appl. Springer, New York, 2009, pp. 157–270.

[Lu21] Jianjie Lu. “Mixed-integer nonlinear modeling and optimization of designing decentralized energy supply networks”. PhD dissertation. Universität Konstanz, 2021.