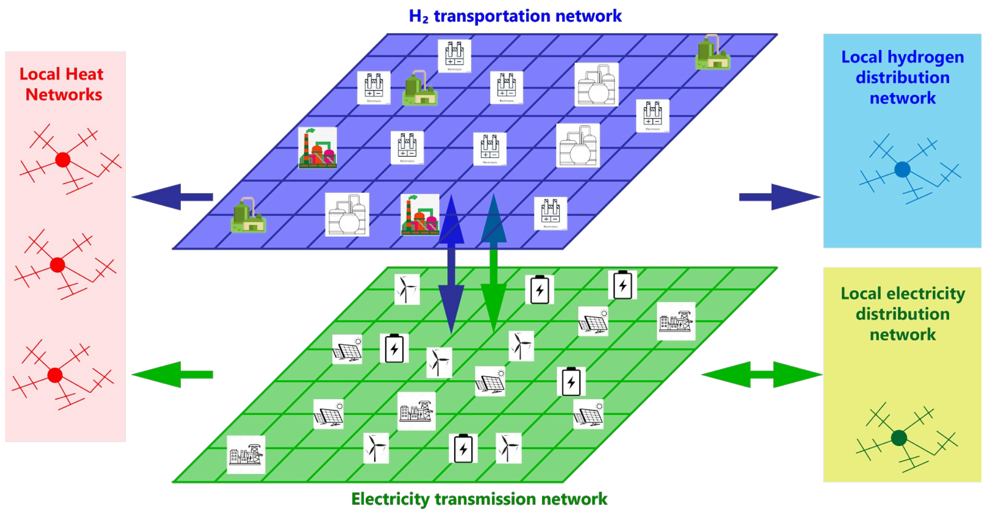
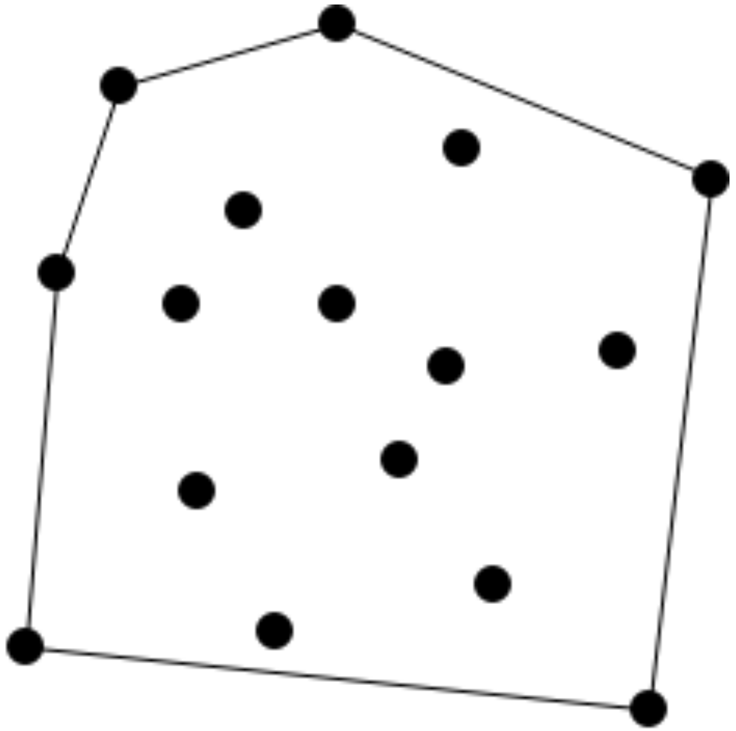
Improving the accuracy of LP storage models: a systematic approach using XPORTA.jl

Maaike B. Elgersma

TU Delft


H₂ transportation network
Local Heat Networks
Local hydrogen distribution network
Local electricity distribution network
Electricity transmission network

&



TU Delft

Contents

1. Short introduction of myself
2. Optimizing storage operation
3. Why tight formulation?
4. Finding the convex hull using Julia (PORTA)
5. Proving the convex hull
6. Results
7. Conclusions

TUDelft

1. Short introduction of myself

- PhD student at TU Delft
- Electrical engineering, mathematics & computer science
  - Algorithmics and Optimization
- Part of NextGenOpt project
- *Improve scalability and accuracy of large-scale energy system optimization models*

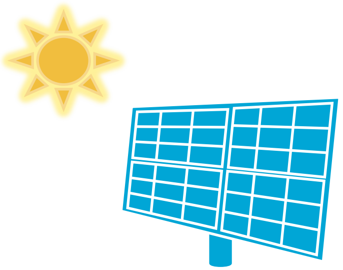
TU Delft

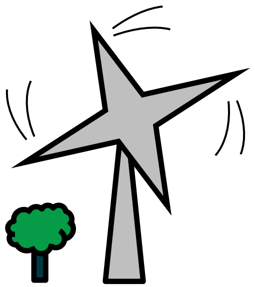
TNO innovation for life

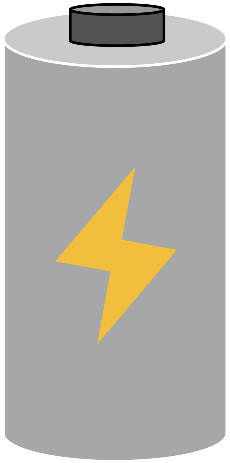
netherlands eScience center

2. Optimizing storage operation

- Goal: Integrate renewable energy systems
- Problem: Production dependent on weather conditions
- → varying production
- Solution: storage
- Including reserves: option to upscale or downscale
- How to operate optimally?

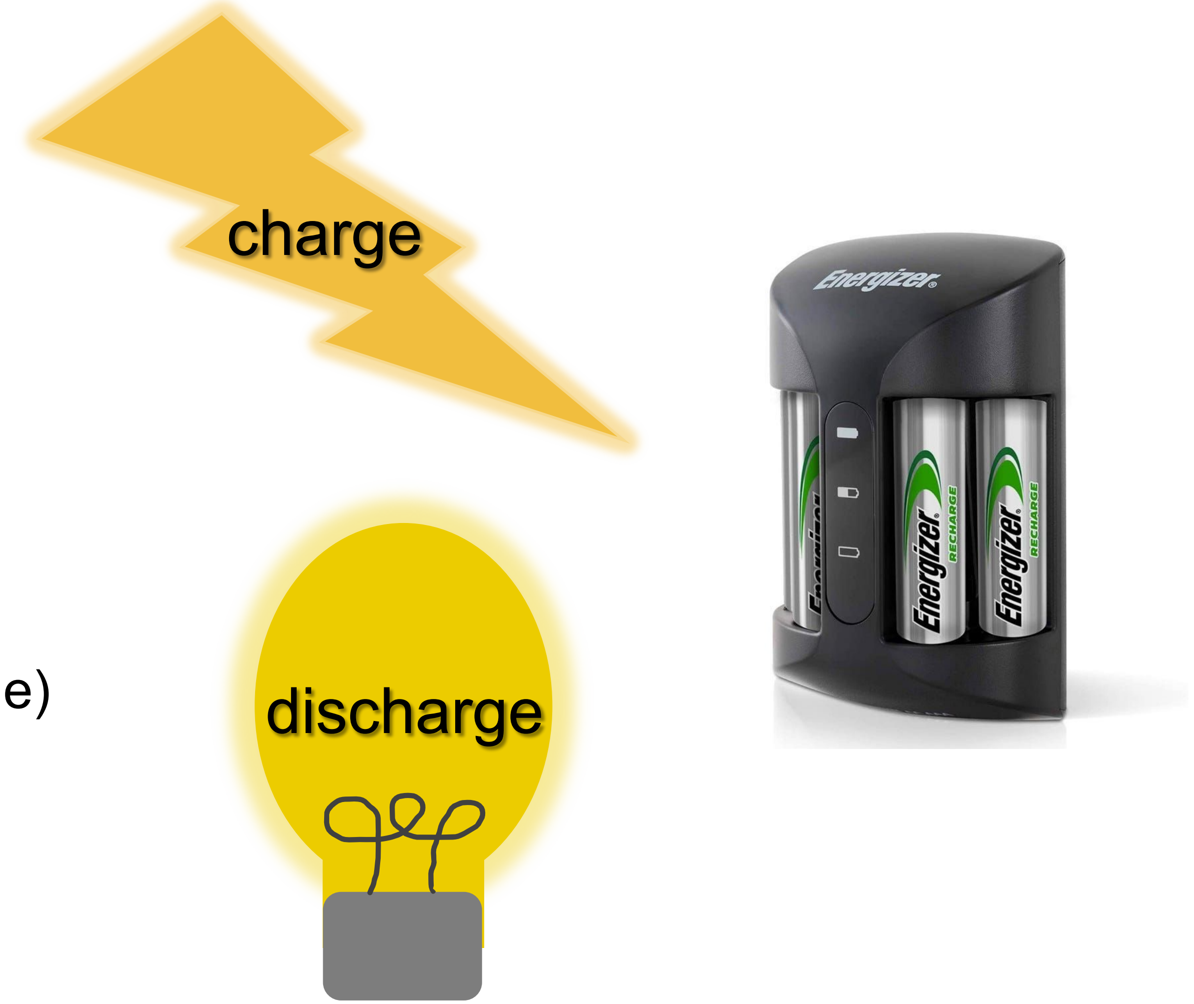






2. Optimizing storage operation

- MILP formulation for storage operation
- Charge **<u>or</u>** discharge
- → Binary decision variable $\delta_t \in \{0,1\}$
- Constraints:
  - charging level: min and max
  - (dis)charge: max per time period
  - reserves: max per time period
- Objective: *minimize operation costs* (example)



3. Why tight formulation?

Problems:

- Storage MILP integrated in large energy system model → very long runtime
- Potential solution: solve relaxed MILP
- But: simultaneous charging and discharging might occur

Previous research:

- Include pre-contingency operating costs [1]
- Roundtrip efficiency < 1 [2]
- Counterexamples show: does not work [3]

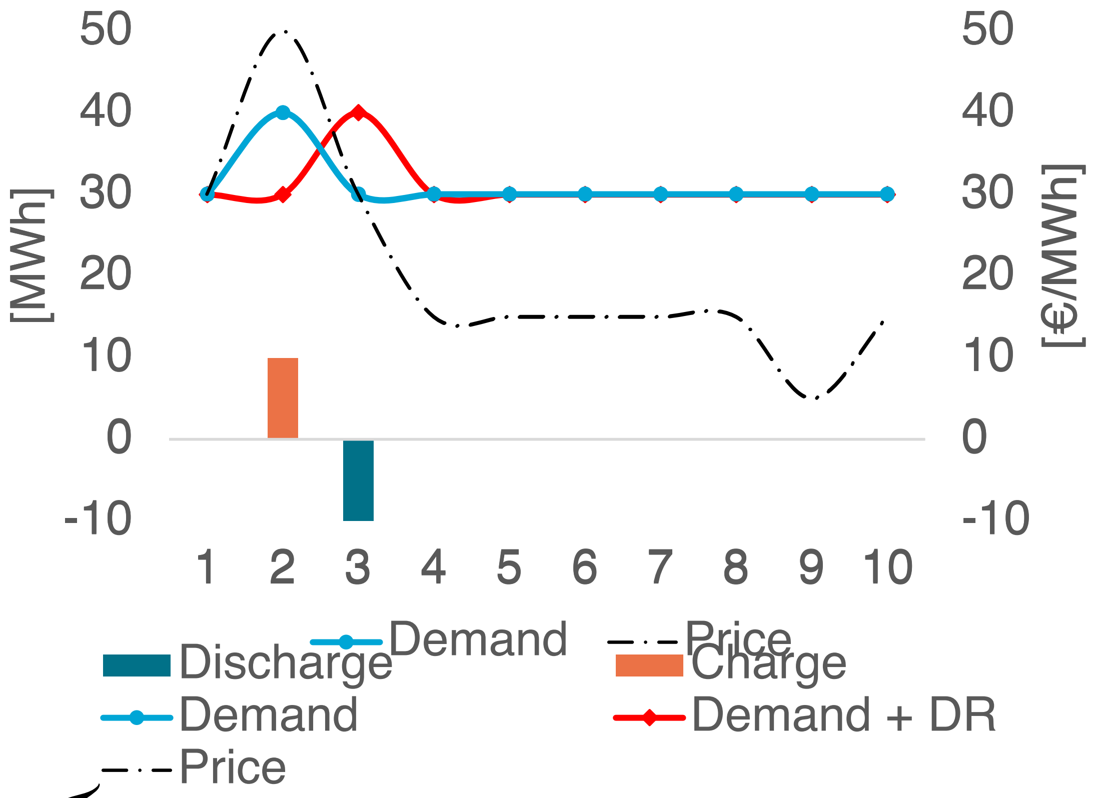
	With z^{S}_{st}		Without z^{S}_{st}	
	Hour		Hour	
	1	2	1	2
p^{G}_{1t} (MW)	12.3	27.3	13.8	28.8
p^{G}_{2t} (MW)	0.0	2.4	0.0	0.0
$p^{\mathrm{S,C}}_{st}$ (MW)	2.3	0.0	5.8	0.0
$p^{\mathrm{S,D}}_{st}$ (MW)	0.0	6.3	2.0	7.2
e^{S}_{st} (MWh)	12.0	5.0	13.0	5.0
MP ($/MWh)	2.6	3.4	−1.0	7.0
Total cost ($)	173.2		130.3	

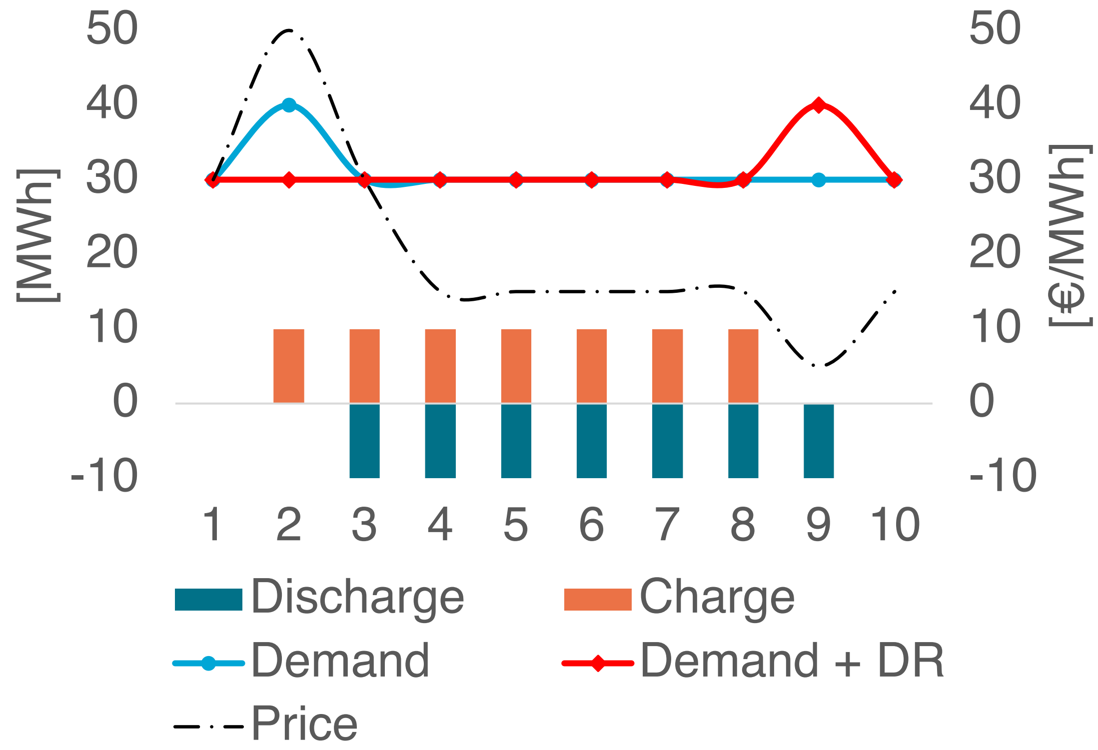
Source: [3]

[1] Z. Li, Q. Guo, H. Sun, and J. Wang, “Sufficient conditions for exact relaxation of complementarity constraints for storage-concerned economic dispatch,” IEEE Trans. Power Syst., vol. 31, no. 2, pp. 1653–1654, Mar. 2016.
[2] B. Zhao, A. J. Conejo, and R. Sioshansi, “Using electrical energy storage to mitigate natural gas-supply shortages,” IEEE Trans. Power Syst., vol. 33, no. 6, pp. 7076–7086, Nov. 2018.
[3] J. M. Arroyo, L. Baringo, A. Baringo, R. Bolanos, N. Alguacil, and N. G. Cobos, “On the Use of a Convex Model for Bulk Storage in MIP-Based Power System Operation and Planning,” IEEE Transactions on Power Systems, vol. 35, no. 6, pp. 4964–4967, Nov. 2020.

3. Why tight formulation?

When does simultaneous charging and discharging occur?

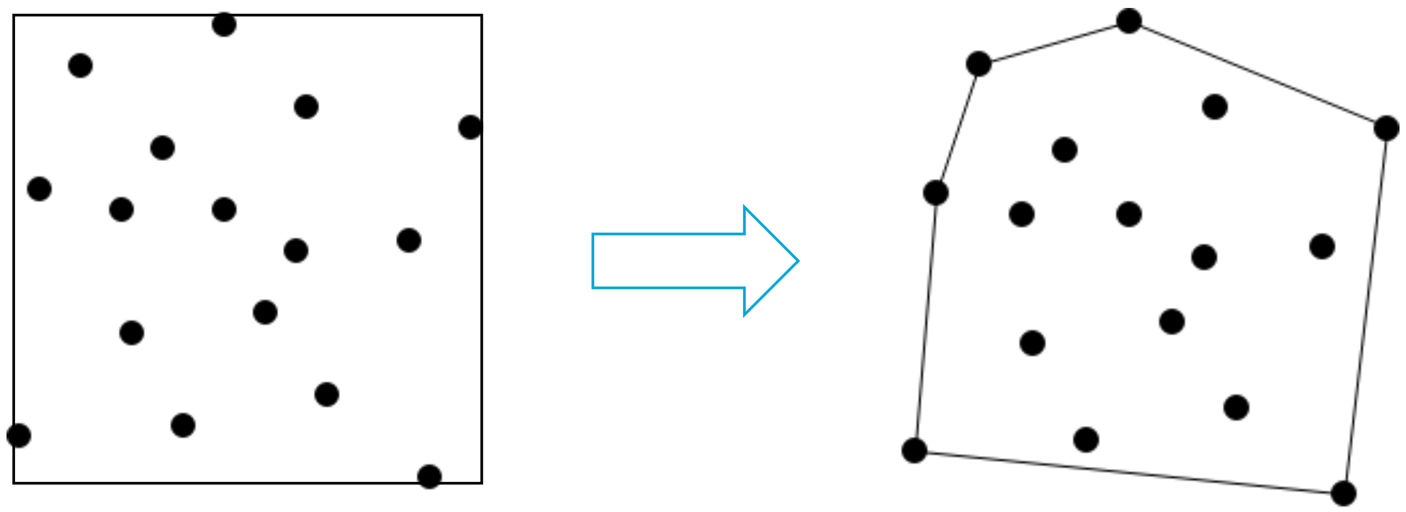




TU Delft

3. Why tight formulation?

- Tighter formulation → MILP solves faster!
- Convex hull: solve relaxed MILP → integer solution!
  - → no simultaneous charging and discharging



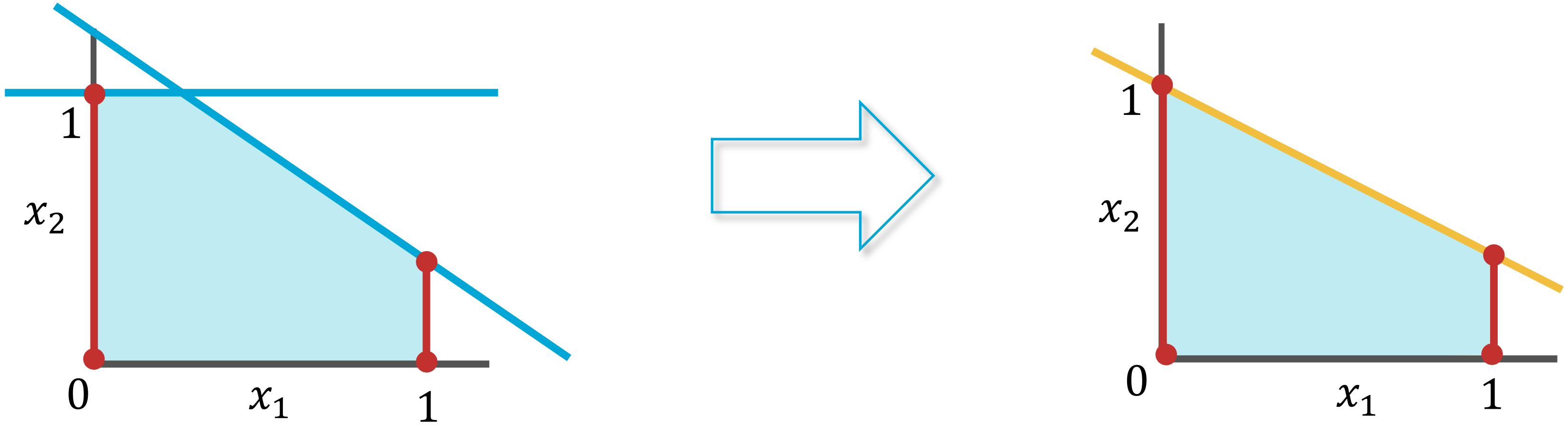
4. Finding the convex hull using Julia (PORTA)

What is PORTA?

- POlyhedron Representation Transformation Algorithm
- Software for analyzing polytopes and polyhedra (https://porta.zib.de/)
- Julia wrapper: **`XPORTA.jl`** (https://github.com/JuliaPolyhedra/XPORTA.jl)
- *Extra recommendation, input from audience: https://github.com/JuliaPolyhedra/Polyhedra.jl*
- **`traf`** function:
  - System of linear (in)equalities → set of points
  - Set of points → system of linear (in)equalities

4. Finding the convex hull using Julia (PORTA)

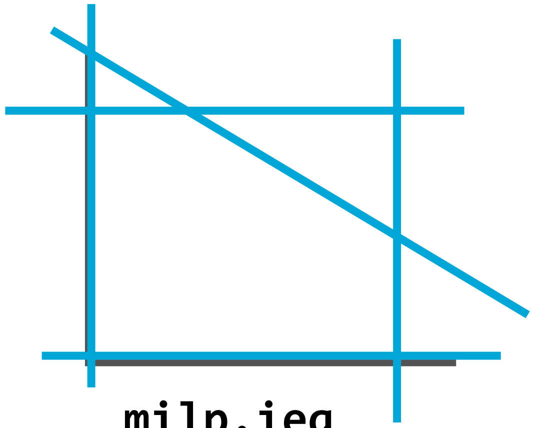
Showcase example:



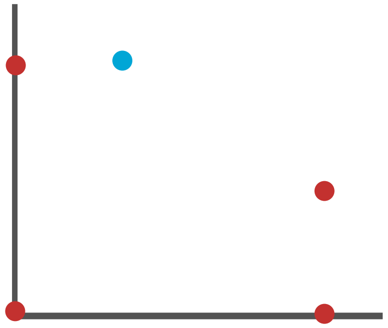
4. Finding the convex hull using Julia (PORTA)

Finding convex hull using PORTA:

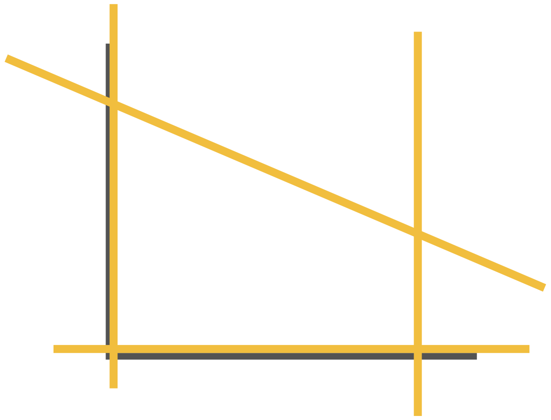
1. Write constraints of original MILP in text file (→ **`milp.ieq`**)
   - For specific format: see guidelines
2. **`traf milp.ieq`** (→ **`milp.ieq.poi`**)
   - Open this file and remove non-integer points
3. **`traf milp.ieq.poi`** (→ **`milp.ieq.poi.ieq`**)



`milp.ieq`



`milp.ieq.poi`



`milp.ieq.poi.ieq`

4. Finding the convex hull using Julia (PORTA)

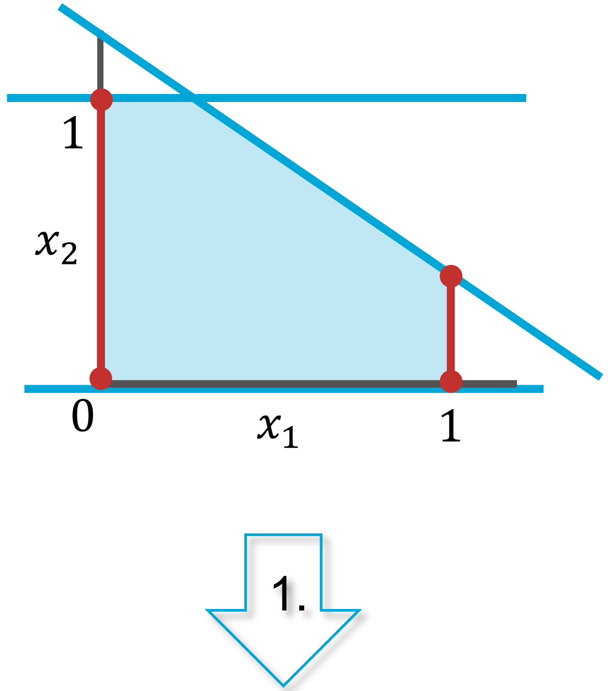
Note: method only works if…

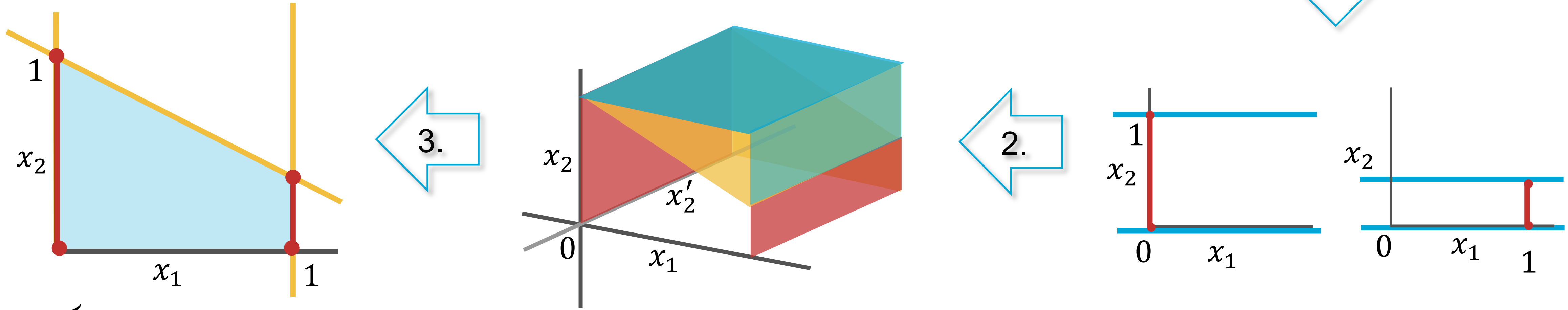
- Problem size/complexity is limited
- Parameter values are known
    - If not known: try for many different values & combinations…

5. Proving the convex hull

Sketch of proof:

1. Write disjunctive set of constraints
2. Write convex hull of these sets ($x_1 = 0$ & $x_2 = 1$) [4]
3. Reduce dimensionality by Fourier-Motzkin elimination
   - Proof that all other constraints are redundant





TU Delft

[4] E. Balas, “Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems,” SIAM. J. on Algebraic and Discrete Methods, vol. 6, no. 3, pp. 466–486, Jul. 1985.

6. Results

Convex hull for storage operation: (1 time period)

8 Basic formulation of 8 constraints

(same as original MILP)

If $a \leq b$:
$x \leq b$ redundant by $x \leq a$

10 10 extra constraints for some parameter values

→ Tried many parameter combinations in PORTA…

76 Proven redundancy 76 times…

2 case studies in JuMP → it works!

TUDelft

7. Conclusions

- PORTA can be very useful tool!
  - But can be hard to get into…
- Proof involves much hardcore mathematics
  - Especially proving redundancy…
  - PORTA can help here!
- Paper on the way... incl. full proof!
  - m.b.elgersma@tudelft.nl
  - Mailing list
- Model will be implemented to speed up large-scale model
  - Come see my poster!

TUDelft

Questions?

Maaike B. Elgersma

m.b.elgersma@tudelft.nl

TU Delft

Bonus slide

Code example using XPORTA.jl:

```julia
import XPORTA

directory = "/Users/mbelgersma/surfdrive/Documents/PhD jaar 1/A Research/A2 Storage paper/code/"
filename = "EmCDRpRmEp.ieq"

# Read the halfspace representation from file (IEQ)
milp_ieq = XPORTA.read_ieq(directory*filename)
# Compute the vertex representation (POI)
milp_poi = XPORTA.traf(milp_ieq)
# Write the vertex representation (IEQ.POI)
XPORTA.write_poi(filename, milp_poi, dir=directory)

# Now open the file and remove the non-integer points

# Read the vertex representation from file (IEQ.POI)
convhull_poi = XPORTA.read_poi(directory*filename*".poi")
# Compute the halfspace representation (IEQ)
convhull_ieq = XPORTA.traf(convhull_poi)
# Write the halfspace respresentation to file (IEQ.POI.IEQ)
XPORTA.write_ieq(filename*".poi", convhull_ieq, dir=directory)

println("Succes!")
```