Optimal Operation of Transient Gas Transport Networks

Kai Hoppmann-Baum

Combinatorial Optimization @ Work 2020

Gas Network Control Problem

General Description

- Optimizing the short-term transient control of large real-world gas transport networks
- "Navigation system" for gas network operators

Source: Open Grid Europe

Problem

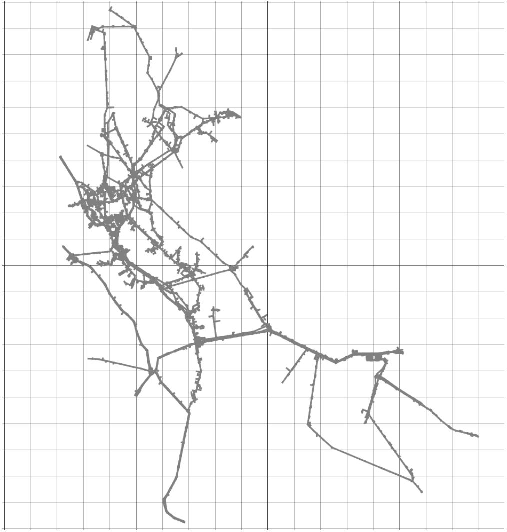
Given

- Network topology
- Initial network state
- Short-term supply/demand and pressure forecast, e.g., 12–24 hours

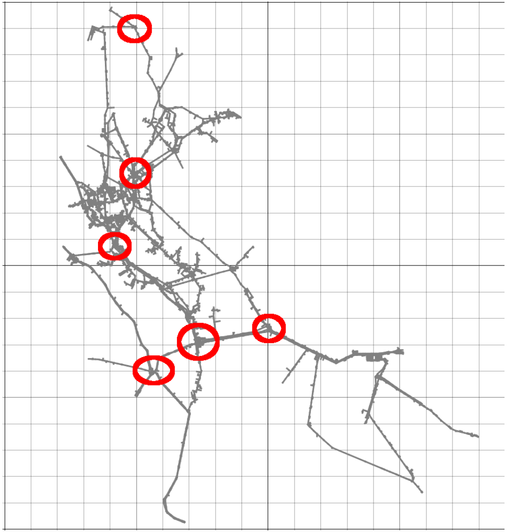
Goal

- Control each element such that the network is operated "best"
- Good control here means: Satisfy all supplies and demands while changing network control as little as possible

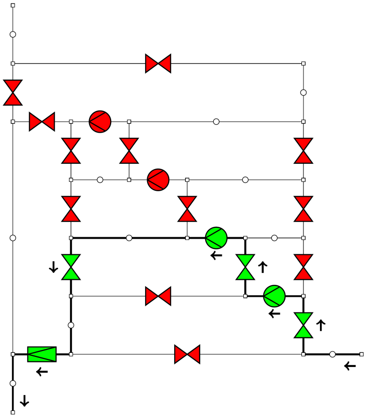
Example Gas Grid



Example Gas Grid - Network Stations



Two Main Sources of Complexity



Combinatorics of Network Stations

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho v)}{\partial x} = 0$$

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial p}{\partial x} + \frac{\partial(\rho v^2)}{\partial x} + \frac{\lambda_a}{2D_a}|v|v\rho + g s_a \rho = 0$$

Transient Gas Flow in Pipelines
Isothermal Euler Equations

KOMPASS - Basic Algorithmic Framework

1. Replace network stations by simplified graph representation

KOMPASS - Basic Algorithmic Framework

CO@Work 2020

1. Replace network stations by simplified graph representation
2. Further network simplifications:

KOMPASS - Basic Algorithmic Framework

1. Replace network stations by simplified graph representation
2. Further network simplifications:
   - Merge pipes (parallel, sequential)

KOMPASS - Basic Algorithmic Framework

1. Replace network stations by simplified graph representation
2. Further network simplifications:
   - Merge pipes (parallel, sequential)
   - Remove distribution network parts

KOMPASS - Basic Algorithmic Framework

1. Replace network stations by simplified graph representation
2. Further network simplifications:
   - Merge pipes (parallel, sequential)
   - Remove distribution network parts
   - ...

KOMPASS - Basic Algorithmic Framework

CO@Work 2020

1. Replace network stations by simplified graph representation
2. Further network simplifications:
   - Merge pipes (parallel, sequential)
   - Remove distribution network parts
   - ...
3. Solve transient operation problem using linearized gas flow equations (Netmodel Algorithm)

KOMPASS - Basic Algorithmic Framework

1. Replace network stations by simplified graph representation
2. Further network simplifications:
   - Merge pipes (parallel, sequential)
   - Remove distribution network parts
   - ...
3. Solve transient operation problem using linearized gas flow equations (Netmodel Algorithm)
4. Retrieve pressure value and flow value time-series for the boundaries nodes of network stations

KOMPASS - Basic Algorithmic Framework

CO@Work 2020

1. Replace network stations by simplified graph representation
2. Further network simplifications:
   - Merge pipes (parallel, sequential)
   - Remove distribution network parts
   - ...
3. Solve transient operation problem using linearized gas flow equations (Netmodel Algorithm)
4. Retrieve pressure value and flow value time-series for the boundaries nodes of network stations
5. Solve transient operation problem for original network stations (Station Model Algorithm)

KOMPASS - Basic Algorithmic Framework

1. Replace network stations by simplified graph representation
2. Further network simplifications:
   - Merge pipes (parallel, sequential)
   - Remove distribution network parts
   - ...
3. Solve transient operation problem using linearized gas flow equations (Netmodel Algorithm)
4. Retrieve pressure value and flow value time-series for the boundaries nodes of network stations
5. Solve transient operation problem for original network stations (Station Model Algorithm)
6. Retrieve control suggestions for dispatchers, i.e., operation modes, target values,...

Outside Network Stations - Pipelines

Gasflow in a pipe (u, v) between timesteps t_i and t_{i+1} can be described by

$$\frac{p_{u,t_{i+1}} - p_{u,t_i}}{2} + \frac{p_{v,t_{i+1}} - p_{v,t_i}}{2} + \frac{R_s\, T\, z\, \Delta t}{L\, A} \left(q_{v,t_{i+1}} - q_{u,t_{i+1}}\right) = 0$$

$$\frac{\lambda\, R_s\, T\, z\, L}{4\, A^2\, D} \left(\frac{\left|q_{u,t_i}\right| q_{u,t_i}}{p_{u,t_i}} + \frac{\left|q_{v,t_i}\right| q_{v,t_i}}{p_{v,t_i}} \right)$$

$$+ \frac{g\, s\, L}{2\, R_s\, T\, z} \left(p_{u,t_i} + p_{v,t_i}\right) + p_{v,t_i} - p_{u,t_i} = 0$$

Outside Network Stations - Pipelines

Gasflow in a pipe (u, v) between timesteps t_i and t_{i+1} can be described by

$$\frac{p_{u,t_{i+1}} - p_{u,t_i}}{2} + \frac{p_{v,t_{i+1}} - p_{v,t_i}}{2} + \frac{R_s\, T\, z\, \Delta t}{L\, A} \left(q_{v,t_{i+1}} - q_{u,t_{i+1}}\right) = 0$$

$$\frac{\lambda\, R_s\, T\, z\, L}{4\, A^2\, D} \left(\frac{|q_{u,t_i}|\, q_{u,t_i}}{p_{u,t_i}} + \frac{|q_{v,t_i}|\, q_{v,t_i}}{p_{v,t_i}} \right)$$

$$+ \frac{g\, s\, L}{2\, R_s\, T\, z} \left(p_{u,t_i} + p_{v,t_i}\right) + p_{v,t_i} - p_{u,t_i} = 0$$

Outside Network Stations - Pipelines

Gasflow in a pipe (u, v) between timesteps t_i and t_{i+1} can be described by

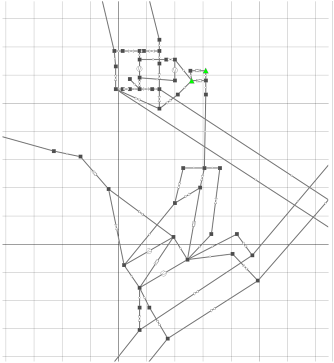
$$\frac{p_{u,t_{i+1}} - p_{u,t_i}}{2} + \frac{p_{v,t_{i+1}} - p_{v,t_i}}{2} + \frac{R_s\, T\, z\, \Delta t}{L\, A} \left(q_{v,t_{i+1}} - q_{u,t_{i+1}}\right) = 0$$

Fixing absolute velocity:

$$\frac{\lambda\, L}{4\, A\, D} \left(|v_{u,0}|\, q_{u,t_i} + |v_{v,0}|\, q_{v,t_i} \right) + \frac{g\, s\, L}{2\, R_s\, T\, z} \left(p_{u,t_i} + p_{v,t_i}\right) + p_{v,t_i} - p_{u,t_i} = 0$$

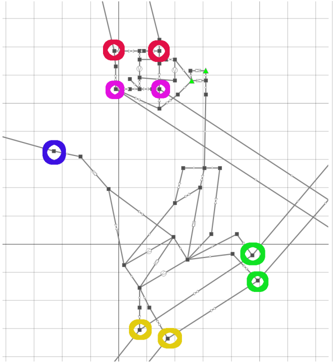
Simplifying Network Stations

- Network stations are bounded by fence nodes



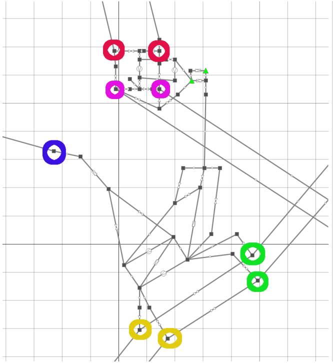
Simplifying Network Stations

- Network stations are bounded by fence nodes



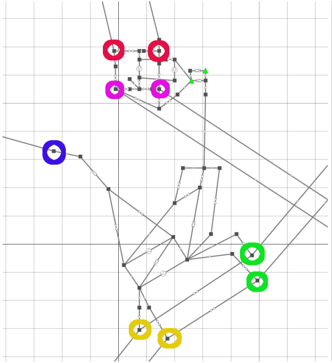
Simplifying Network Stations

- Network stations are bounded by fence nodes
- Elements between fence nodes are removed



Simplifying Network Stations

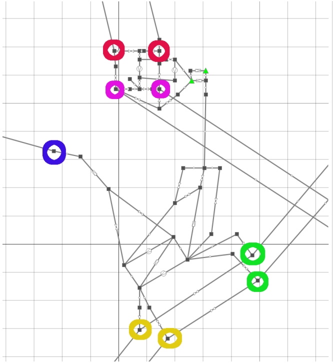
- Network stations are bounded by fence nodes
- Elements between fence nodes are removed
- Fence nodes with similar "behaviour" are grouped into fence groups



Simplifying Network Stations

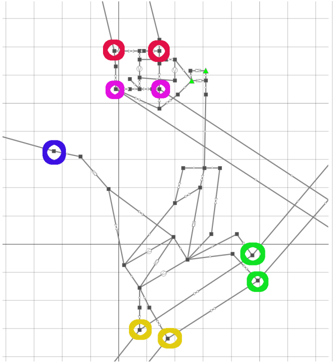
CO@Work 2020

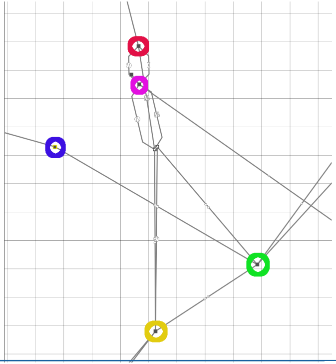
- Network stations are bounded by fence nodes
- Elements between fence nodes are removed
- Fence nodes with similar “behaviour” are grouped into fence groups
- Nodes in a fence group are merged into a single nodes



Simplifying Network Stations

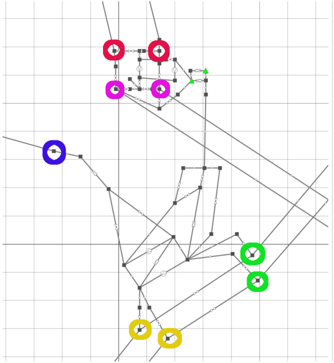
- Network stations are bounded by fence nodes
- Elements between fence nodes are removed
- Fence nodes with similar "behaviour" are grouped into fence groups
- Nodes in a fence group are merged into a single nodes

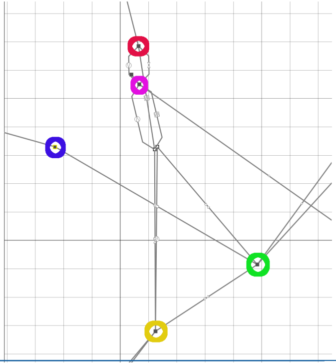




Simplifying Network Stations

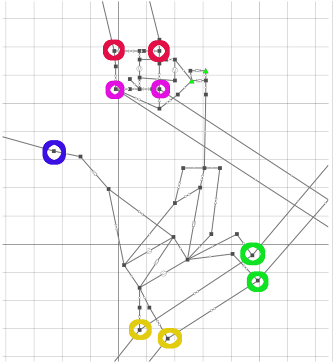
- Network stations are bounded by fence nodes
- Elements between fence nodes are removed
- Fence nodes with similar “behaviour” are grouped into fence groups
- Nodes in a fence group are merged into a single nodes
- Auxiliary nodes (for modelling purposes) may be introduced

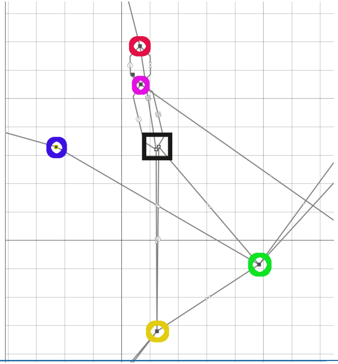




Simplifying Network Stations

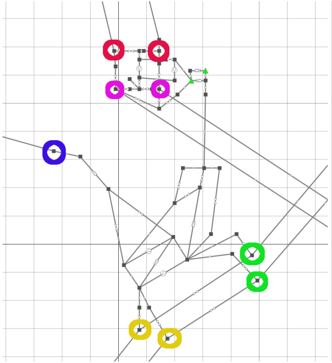
- Network stations are bounded by fence nodes
- Elements between fence nodes are removed
- Fence nodes with similar “behaviour” are grouped into fence groups
- Nodes in a fence group are merged into a single nodes
- Auxiliary nodes (for modelling purposes) may be introduced

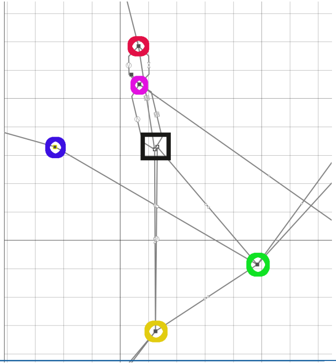




Simplifying Network Stations

- Network stations are bounded by fence nodes
- Elements between fence nodes are removed
- Fence nodes with similar "behaviour" are grouped into fence groups
- Nodes in a fence group are merged into a single nodes
- Auxiliary nodes (for modelling purposes) may be introduced
- Auxiliary links represent the capabilities of a network station



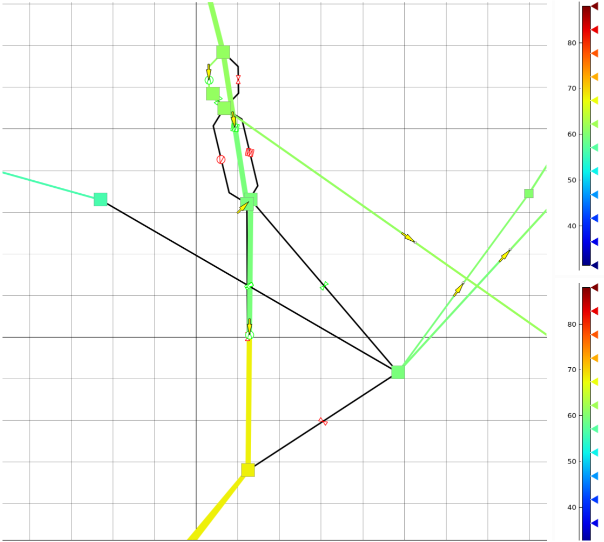


Flow Directions and Simple States

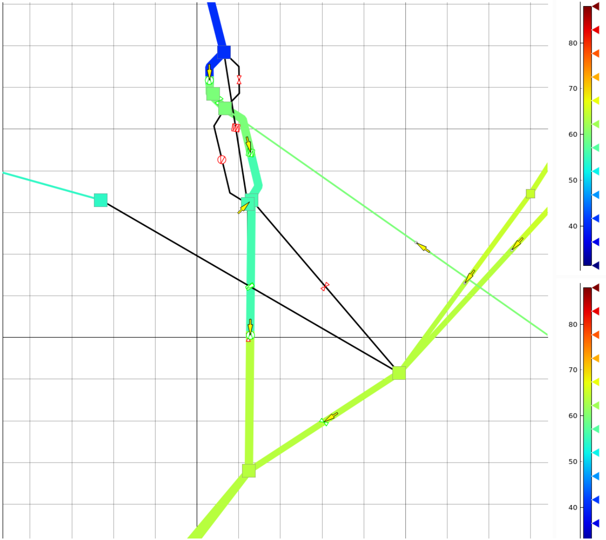
For each network station (V, A) we are given

- Flow directions $\mathcal{F} \subseteq \mathcal{P}(V) \times \mathcal{P}(V)$ with $f = (f^+, f^-) \in \mathcal{F}$
- Simple states $\mathcal{S} \subseteq \mathcal{P}(\mathcal{F}) \times \mathcal{P}(A) \times \mathcal{P}(A)$ with $s = (s_f, s_a^{on}, s_a^{off}) \in \mathcal{S}$

Example I



Example II



Flow Directions and Simple States

For each network station (V, A) we are given

- Flow directions $\mathcal{F} \subseteq \mathcal{P}(V) \times \mathcal{P}(V_i)$ (example: (f^+, f^-))
- Simple states $\mathcal{S} \subseteq \mathcal{P}(\mathcal{F}) \times \mathcal{P}(A) \times \mathcal{P}(A)$ (example: $(s_f, s_a^{on}, s_a^{off})$)
- $x_{f,t} \in \{0,1\}$ for flow direction $f \in \mathcal{F}$ and time step $t \in T$
- $x_{s,t} \in \{0,1\}$ for simple state $s \in \mathcal{S}$ and time step $t \in T$
- $x_{a,t} \in \{0,1\}$ for artificial arc $a \in A$ and time step $t \in T$

$$
\begin{aligned}
\sum\nolimits_{f \in \mathcal{F}} x_{f,t} &= 1 && \forall t \in T \\
\sum\nolimits_{f \in s_f} x_{f,t} &\geq x_{s,t} && \forall s \in \mathcal{S}, \forall t \in T \\
\sum\nolimits_{s \in \mathcal{S}} x_{s,t} &= 1 && \forall t \in T \\
x_{s,t} &\leq x_{a,t} && \forall s \in S, \forall a \in s_a^{on}, \forall t \in T \\
1 - x_{s,t} &\geq x_{a,t} && \forall s \in S, \forall a \in s_a^{off}, \forall t \in T
\end{aligned}
$$

... additional flow direction related constraints ...

Shortcuts

For a shortcut $a = (u, v)$ and each $t \in T$:

Not Active ($x_{a,t} = 0$):

- Decoupled pressure values
- No flow allowed

Active ($x_{a,t} = 1$):

- Coupled pressure values
- Bidirectional flow up to an amount of $\overline{q}_a$ (Big-M).

$$
\begin{aligned}
p_{u,t} - p_{v,t} &\leq (1 - x_{a,t})(\overline{p}_v - \underline{p}_u) \\
p_{u,t} - p_{v,t} &\geq (1 - x_{a,t})(\underline{p}_v - \overline{p}_u) \\
q^{\rightarrow}_{a,t} &\leq x_{a,t}\overline{q}_a \\
q^{\leftarrow}_{a,t} &\leq x_{a,t}\overline{q}_a.
\end{aligned}
$$

Regulating Arcs

For a regulating arc $a = (u, v)$ and each $t \in T$:

Not Active ($x_{a,t} = 0$):

- Decoupled pressure values
- No flow allowed

Active ($x_{a,t} = 1$):

- Pressure at u not smaller than pressure at v
- Unidirectional flow up to an amount of $\overline{q}_a$ (Big-M).

$$p_{u,t} - p_{v,t} \geq (1 - x_{a,t})(\underline{p}_v - \overline{p}_u)$$
$$q^{\rightarrow}_{a,t} \leq x_{a,t}\overline{q}_a.$$

Compressing Arcs

Not Active ($x_{a,t} = 0$):

- No machine assigned
- Decoupled pressure values
- No flow allowed

Active ($x_{a,t} = 1$):

- Assign machines to compressing arc
- Pressure at v not smaller than pressure at u
- Pressure at v at most r_a times greater than $p_{u,0}$
- Flow limited by sum of max flows of assigned machines
- Respect approximated power bound equation

Compressing Arcs - MILP Model

For each machine $i \in M$ and for each timestep $t \in T$ we have

$$\sum_{a \in A: i \in M_a} y^i_{a,t} \leq 1$$

$$y^i_{a,t} \leq x_{a,t}$$

For each compressing arc a and for each timestep $t \in T$ we have

$$q^{\rightarrow}_{a,t} \leq \sum_{i \in M_a} F^i y^i_{a,t}$$

$$r_{a,t} = 1 + \sum_{i \in M_a} (1 - R^i) y^i_{a,t}$$

$$\pi_{a,t} \leq \sum_{i \in M_a} P^i y^i_{a,t}$$

$$p_{u,t} - p_{v,t} \leq (1 - x_{a,t})(\overline{p}_v - \underline{p}_u)$$

$$r_a p_{u,0} - p_{v,t} \geq (1 - x_{a,t})(p_{u,0} - \overline{p}_{v,t})$$

$$\alpha_1 p_{u,t} + \alpha_2 p_{v,t} + \alpha_3 q^{\rightarrow}_{a,t} + \alpha_4 \pi_{a,t} \leq \beta x_{a,t} + (1 - x_{a,t})(\alpha_1 \underline{p}_u + \alpha_2 \overline{p}_v + \alpha_3 \overline{q}_a)$$

$$\alpha_1 p_{u,t} + \alpha_2 p_{v,t} + \alpha_3 q^{\rightarrow}_{a,t} + \alpha_4 \pi_{a,t} \geq \beta x_{a,t} + (1 - x_{a,t})(\alpha_1 \overline{p}_u + \alpha_2 \underline{p}_v + \alpha_4 \overline{\pi}_a)$$

Combined Arcs

Introduce binary variables $x^r_{a,t}, x^c_{a,t} \in \{0,1\}$ indicating whether the arc is regulating or compressing:

$$x^r_{a,t} + x^c_{a,t} = x_{a,t}$$

Not Active ($x_{a,t} = 0$):

- No machine assigned
- Decoupled pressure values
- No flow allowed

Active and regulating ($x_{a,t} = 1$ and $x^r_{a,t} = 1$):

- Like regulating arc

Active and compressing ($x_{a,t} = 1$ and $x^c_{a,t} = 1$):

- Like compressing arc

Coupling Pipelines and Network Stations & Objective

Flow conservation holds at all nodes in the network

$$\sum \text{outgoing flow} - \sum \text{ingoing flow} = b_{v,t}$$

where $b_{v,t} = 0$ for inner nodes, $b_{v,t} \geq 0$ for entries, and $b_{v,t} \leq 0$ for exits.

Coupling Pipelines and Network Stations & Objective

Flow conservation holds at all nodes in the network

$$\sum \text{outgoing flow} - \sum \text{ingoing flow} = b_{v,t}$$

where $b_{v,t} = 0$ for inner nodes, $b_{v,t} \geq 0$ for entries, and $b_{v,t} \leq 0$ for exits.

The (current) objective of Netmodel-MILP is to minimize the number of

1. flow direction changes,
2. simple state changes,
3. and artificial link switches.

Coupling Pipelines and Network Stations & Objective

Flow conservation holds at all nodes in the network

$$\sum \text{outgoing flow} - \sum \text{ingoing flow} = b_{v,t}$$

where $b_{v,t} = 0$ for inner nodes, $b_{v,t} \geq 0$ for entries, and $b_{v,t} \leq 0$ for exits.

The (current) objective of Netmodel-MILP is to minimize the number of

1. flow direction changes,
2. simple state changes,
3. and artificial link switches.

Currently, we discuss to additionally penalize

- compressor/combined links being active,
- assigning machines,
- power used for compression,
- ...

3-Stage Approach and Solution Smoothening

CO@Work 2020

To avoid infeasibility, we have a 3-Stage Approach

1. Initial MILP

3-Stage Approach and Solution Smoothening

To avoid infeasibility, we have a 3-Stage Approach

1. Initial MILP
2. Stage 1 infeasible $\Rightarrow$ add (expensive) slack on supplies/demands

3-Stage Approach and Solution Smoothening

To avoid infeasibility, we have a 3-Stage Approach

1. Initial MILP
2. Stage 1 infeasible $\Rightarrow$ add (expensive) slack on supplies/demands
3. Stage 2 infeasible $\Rightarrow$ add (highly expensive) slack on pressure bounds

3-Stage Approach and Solution Smoothening

To avoid infeasibility, we have a 3-Stage Approach

1. Initial MILP
2. Stage 1 infeasible $\Rightarrow$ add (expensive) slack on supplies/demands
3. Stage 2 infeasible $\Rightarrow$ add (highly expensive) slack on pressure bounds

The last MILP should always admit a feasible solution.

3-Stage Approach and Solution Smoothening

To avoid infeasibility, we have a 3-Stage Approach

1. Initial MILP
2. Stage 1 infeasible $\Rightarrow$ add (expensive) slack on supplies/demands
3. Stage 2 infeasible $\Rightarrow$ add (highly expensive) slack on pressure bounds

The last MILP should always admit a feasible solution. It should...

3-Stage Approach and Solution Smoothening

To avoid infeasibility, we have a 3-Stage Approach

1. Initial MILP
2. Stage 1 infeasible $\Rightarrow$ add (expensive) slack on supplies/demands
3. Stage 2 infeasible $\Rightarrow$ add (highly expensive) slack on pressure bounds

The last MILP should always admit a feasible solution. It should...

For a Netmodel-MILP solution we solve a smoothening given by

3-Stage Approach and Solution Smoothening

To avoid infeasibility, we have a 3-Stage Approach

1. Initial MILP
2. Stage 1 infeasible ⇒ add (expensive) slack on supplies/demands
3. Stage 2 infeasible ⇒ add (highly expensive) slack on pressure bounds

The last MILP should always admit a feasible solution. It should...

For a Netmodel-MILP solution we solve a smoothening given by

1. Fix all binary variables in Netmodel-MILP

3-Stage Approach and Solution Smoothening

To avoid infeasibility, we have a 3-Stage Approach

1. Initial MILP
2. Stage 1 infeasible $\Rightarrow$ add (expensive) slack on supplies/demands
3. Stage 2 infeasible $\Rightarrow$ add (highly expensive) slack on pressure bounds

The last MILP should always admit a feasible solution. It should...

For a Netmodel-MILP solution we solve a smoothening given by

1. Fix all binary variables in Netmodel-MILP
2. Introduce variables and constraints accounting for pressure and flow differences at boundaries between t and $t - i$

3-Stage Approach and Solution Smoothening

To avoid infeasibility, we have a 3-Stage Approach

1. Initial MILP
2. Stage 1 infeasible ⇒ add (expensive) slack on supplies/demands
3. Stage 2 infeasible ⇒ add (highly expensive) slack on pressure bounds

The last MILP should always admit a feasible solution. It should...

For a Netmodel-MILP solution we solve a smoothening given by

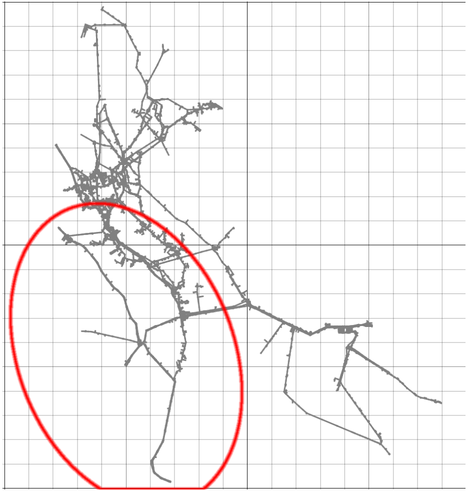
1. Fix all binary variables in Netmodel-MILP
2. Introduce variables and constraints accounting for pressure and flow differences at boundaries between t and $t - i$
3. Minimize the sum of these variables

Netmodel-Algorithm

Solve MILP
if MILP is infeasible **then**
    Add slack on supply/demands and resolve
    if MILP is infeasible **then**
        Add slack on pressure bounds and resolve
$\text{sol}_0 \leftarrow$ smoothened solution of MILP

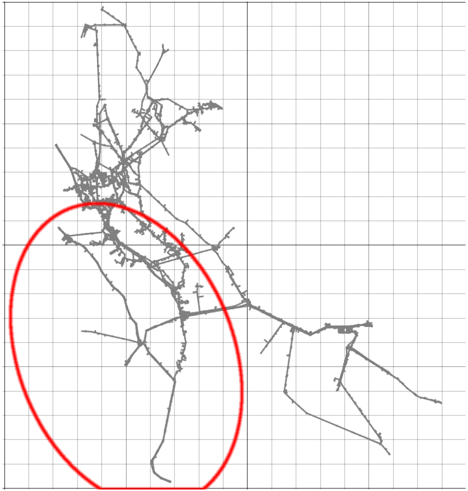
Future Work and Challenges

- Automatization of simplified graph representation



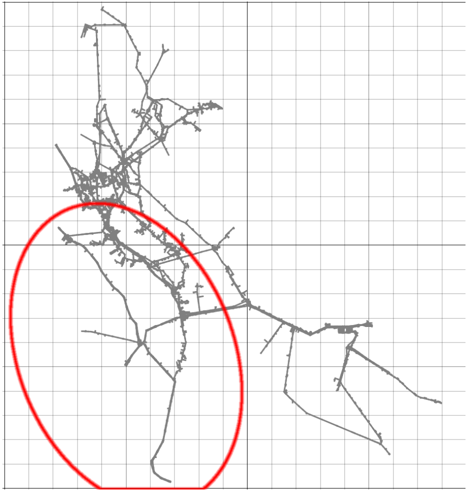
Future Work and Challenges

- Automatization of simplified graph representation
- Solution for non-linear Momentum Equations



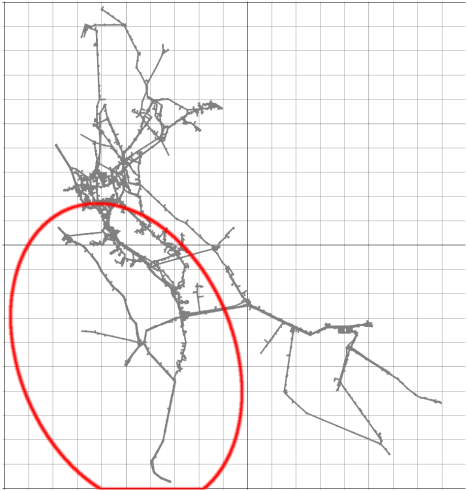
Future Work and Challenges

- Automatization of simplified graph representation
- Solution for non-linear Momentum Equations
- Feedback from Station Model to Netmodel



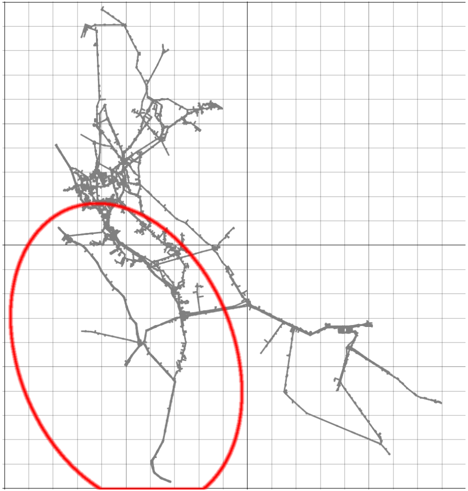
Future Work and Challenges

- Automatization of simplified graph representation
- Solution for non-linear Momentum Equations
- Feedback from Station Model to Netmodel
- More realistic element modelling



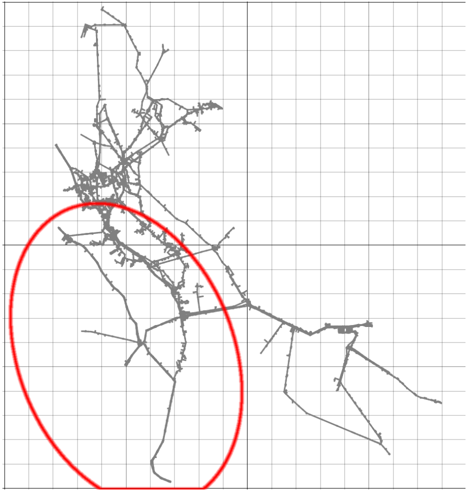
Future Work and Challenges

- Automatization of simplified graph representation
- Solution for non-linear Momentum Equations
- Feedback from Station Model to Netmodel
- More realistic element modelling
  - Air temperature dependent compression power bound



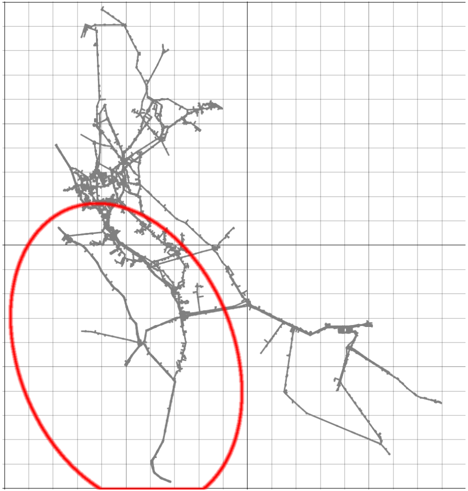
Future Work and Challenges

- Automatization of simplified graph representation
- Solution for non-linear Momentum Equations
- Feedback from Station Model to Netmodel
- More realistic element modelling
  - Air temperature dependent compression power bound
  - Semi-fixed elements



Future Work and Challenges

- Automatization of simplified graph representation
- Solution for non-linear Momentum Equations
- Feedback from Station Model to Netmodel
- More realistic element modelling
  - Air temperature dependent compression power bound
  - Semi-fixed elements
  - Single special network elements



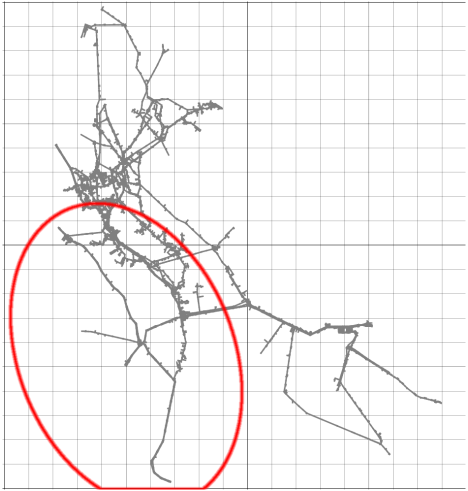
Future Work and Challenges

- Automatization of simplified graph representation
- Solution for non-linear Momentum Equations
- Feedback from Station Model to Netmodel
- More realistic element modelling
  - Air temperature dependent compression power bound
  - Semi-fixed elements
  - Single special network elements
  - Reduce simplifications made in the Netmodel

Future Work and Challenges

CO@Work 2020

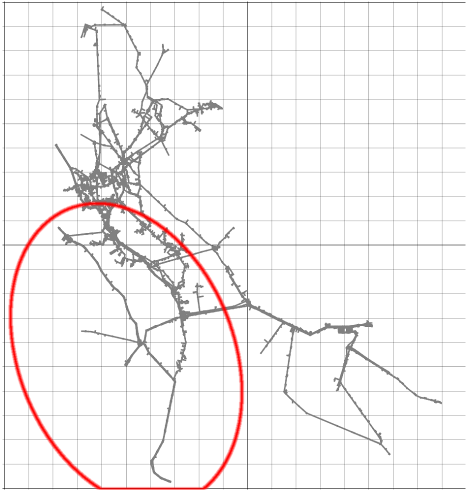
- Automatization of simplified graph representation
- Solution for non-linear Momentum Equations
- Feedback from Station Model to Netmodel
- More realistic element modelling
    - Air temperature dependent compression power bound
    - Semi-fixed elements
    - Single special network elements
    - Reduce simplifications made in the Netmodel
- Stable solutions over sequential runs



Future Work and Challenges

CO@Work 2020

- Automatization of simplified graph representation
- Solution for non-linear Momentum Equations
- Feedback from Station Model to Netmodel
- More realistic element modelling
  - Air temperature dependent compression power bound
  - Semi-fixed elements
  - Single special network elements
  - Reduce simplifications made in the Netmodel
- Stable solutions over sequential runs
- Increase network size



The Last Slide

Thanks for watching!

kai.hoppmann@zib.de hoppmannkai

Combinatorial Optimization @ Work 2020