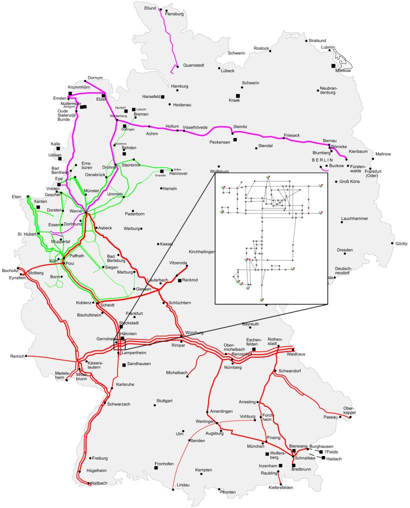
Learning Control Decisions in Gas Networks

Mark Turner

Combinatorial Optimization @ Work 2020

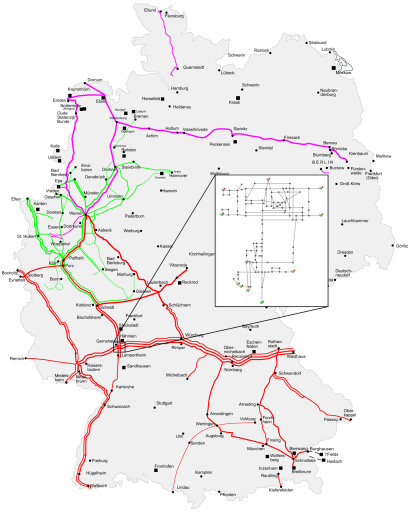
Exact control decisions of Network Stations

- Listen to previous talks
  - Provides descriptions of individual gas network elements
  - Provides background into the derivation of this problem



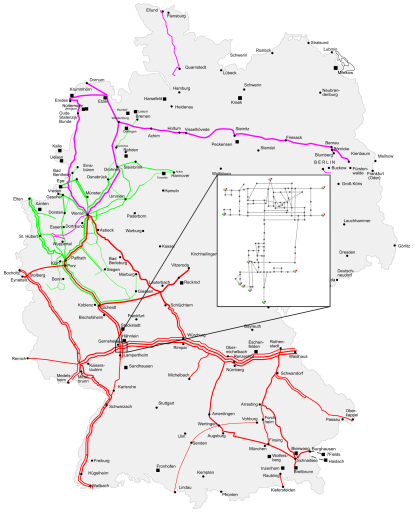
Exact control decisions of Network Stations

- Listen to previous talks
- Problem Setting (Important sub-networks inside larger network)
  - Focus on Network Stations
  - Network Stations contain all heavy machinery of the entire network
  - Network stations are commonly the intersection points between large transportation pipelines



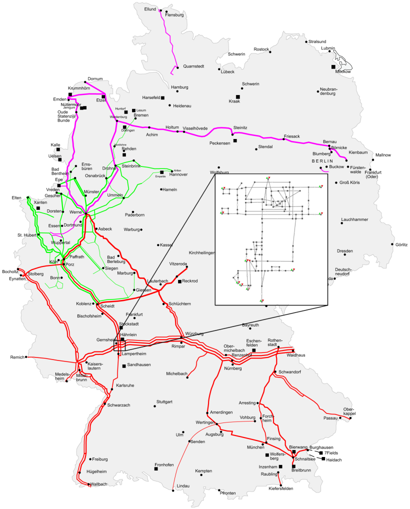
Exact control decisions of Network Stations

- Listen to previous talks
- Problem Setting (Important sub-networks inside larger network)
- Problem Aim (Control Decisions)
  - Find control decisions for all network elements
  - Ensure these control decisions are safe and realisable
  - Make these control decisions as ‘stable’ as possible
  - Control decisions are made at each time-step over a time-horizon $\{0, ..., |T|\}$.



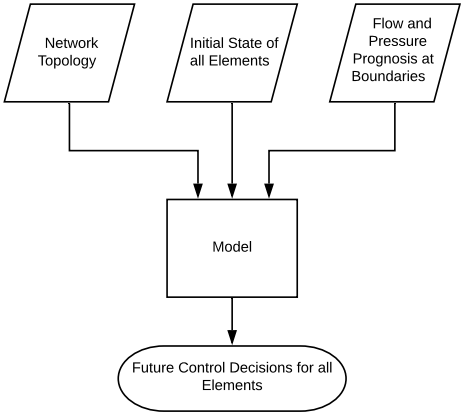
Exact control decisions of Network Stations

- Listen to previous talks
- Problem Setting (Important sub-networks inside larger network)
- Problem Aim (Control Decisions)
- Two Different Approaches:
  - (Model driven) Rolling Horizon approach
  - (Data driven) Machine Learning approach



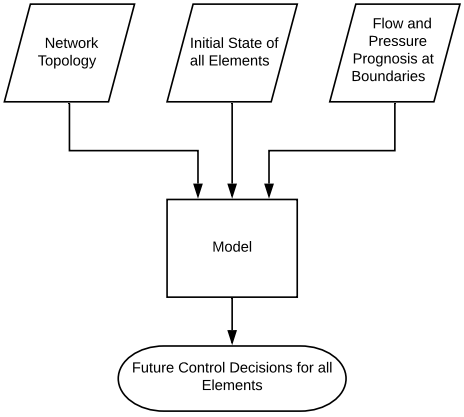
Problem Input and Output

- Input:
  - Network Topology - Complete individual element descriptions and connectedness
  - Initial State - Starting values for all elements
  - Prognosis - Set of demands we aim to meet



Problem Input and Output

- Input:
  - Network Topology - Complete individual element descriptions and connectedness
  - Initial State - Starting values for all elements
  - Prognosis - Set of demands we aim to meet
- Output (For all future time steps):
  - Valve states (Open / Closed)
  - Compressor states (Active (operating point) / Bypass / Closed)
  - Regulator states (Active (operating point) / Open / Closed)
  - Additional: Flow / Pressure levels throughout the network



Problem Uncertainties and Difficulties

- Organising priorities.

Problem Uncertainties and Difficulties

- Organising priorities.
  - Supply and Demand is met

$$\sum_u \text{flow}_{(u,v)} + \sum_u \text{flow}_{(v,u)} = v_{\text{inflow}} - v_{\text{slack}}$$

$$|v_{\text{slack}}| \leq \epsilon \quad v \in \text{Boundaries}$$

Problem Uncertainties and Difficulties

- Organising priorities.
  - Supply and Demand is met
  - Control Decisions are safe

E.g. Make sure there are well defined pressure limits for all bits of the network, and that your solution respects them:

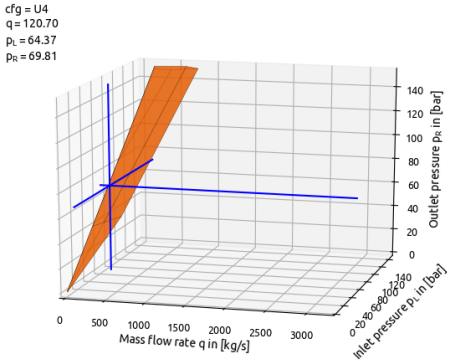
$$\text{LB}(\text{pressure}_v) \leq \text{pressure}_v \leq \text{UB}(\text{pressure}_v)$$

Potentially less well defined requests. E.g. Avoid stressing the network

Problem Uncertainties and Difficulties

- Organising priorities.
  - Supply and Demand is met
  - Control Decisions are safe
  - Control Decisions are stable

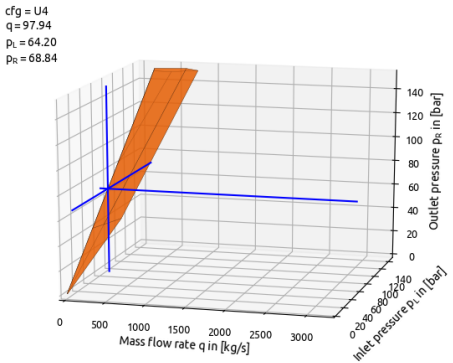
E.g. Make sure that your active point inside of a compressor polytope moves as little as possible



Problem Uncertainties and Difficulties

- Organising priorities.
  - Supply and Demand is met
  - Control Decisions are safe
  - Control Decisions are stable

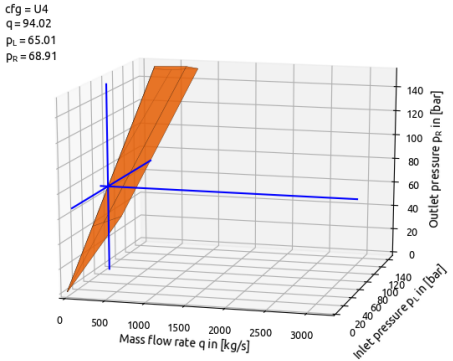
E.g. Make sure that your active point inside of a compressor polytope moves as little as possible



Problem Uncertainties and Difficulties

- Organising priorities.
  - Supply and Demand is met
  - Control Decisions are safe
  - Control Decisions are stable

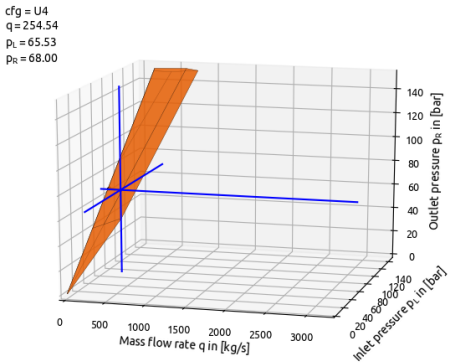
E.g. Make sure that your active point inside of a compressor polytope moves as little as possible



Problem Uncertainties and Difficulties

- Organising priorities.
  - Supply and Demand is met
  - Control Decisions are safe
  - Control Decisions are stable

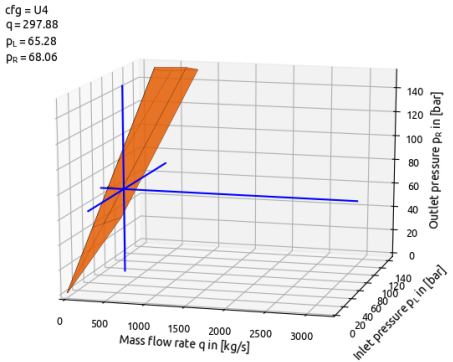
E.g. Make sure that your active point inside of a compressor polytope moves as little as possible



Problem Uncertainties and Difficulties

- Organising priorities.
  - Supply and Demand is met
  - Control Decisions are safe
  - Control Decisions are stable

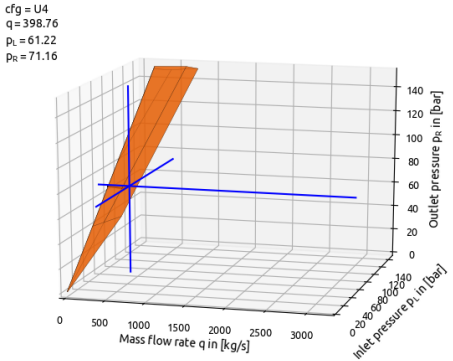
E.g. Make sure that your active point inside of a compressor polytope moves as little as possible



Problem Uncertainties and Difficulties

- Organising priorities.
  - Supply and Demand is met
  - Control Decisions are safe
  - Control Decisions are stable

E.g. Make sure that your active point inside of a compressor polytope moves as little as possible



Problem Uncertainties and Difficulties

- Organising priorities.
  - Supply and Demand is met
  - Control Decisions are safe
  - Control Decisions are stable
  - Weighted Objective vs Multi Level

$$\min_{x,y} \quad Ax + By$$

vs.

$$\min_{x} \quad Ax$$
$$\text{s.t}$$
$$\min_{y} \quad By$$

Problem Uncertainties and Difficulties

- Organising priorities.
- Solution must be output in reasonable time
  - Model will be used in reality.
  - Output in time is necessary

https://xkcd.com/612/

Problem Uncertainties and Difficulties

- Organising priorities.
- Solution must be output in reasonable time
- Model exactness
  - Pipe discretisation
- What assumptions do I make? E.g. Ignore temperature
- Which discretisation technique do I use?
- Do I leave as is, convexify, or linearise my end result?
- See Kai Hoppmann's talk on more information on pipe equations. (We linearise)

Problem Uncertainties and Difficulties

- Organising priorities.
- Solution must be output in reasonable time
- Model exactness
  - Pipe discretisation
  - Compressor polytope

Compressor stations increase gas-pressure in the forward direction. We must choose which configuration of machines to use inside of the compressor station, and the operating point inside of the configuration's associated polytope.

Idea: Disjunctive Formulation

Use case: Efficiently model each compressor configuration polytope s.t we retrieve the active configuration, and the active operating point inside of the polytope.

Problem Uncertainties and Difficulties

- Organising priorities.
- Solution must be output in reasonable time
- Model exactness
  - Pipe discretisation
  - Compressor polytope

$$
\begin{aligned}
m_c \in \{0,1\} \quad & \forall c \in \mathrm{C}: && \text{Active configuration} \\
\mathrm{pl}_c \in \mathbb{R}^+ \quad & \forall c \in \mathrm{C}: && \text{LHS pressure} \\
\mathrm{pr}_c \in \mathbb{R}^+ \quad & \forall c \in \mathrm{C}: && \text{RHS pressure} \\
q_c \in \mathbb{R}^+ \quad & \forall c \in \mathrm{C}: && \text{Flow}
\end{aligned}
$$

$$
\sum_c m_c = 1 \quad \mathrm{pl} = \sum_c \mathrm{pl}_c
$$

$$
\mathrm{pr} = \sum_c \mathrm{pr}_c \quad q = \sum_c q_c
$$

$$
LB(\mathrm{pl}_c) m_c \leq \mathrm{pl}_c \leq UB(\mathrm{pl}_c) m_c
$$

$$
LB(\mathrm{pr}_c) m_c \leq \mathrm{pr}_c \leq UB(\mathrm{pr}_c) m_c
$$

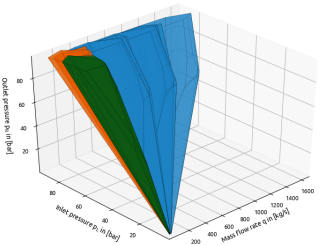
$$
LB(q_c) m_c \leq q_c \leq UB(q_c) m_c
$$

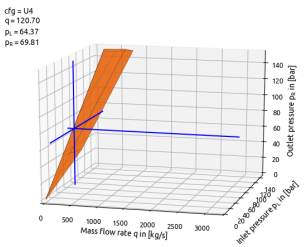
$$
w \cdot \mathrm{pl}_c + x \cdot \mathrm{pr}_c + y \cdot q_c + z \cdot m_c \leq 0 \quad \forall c \in \mathrm{C}
$$

$$
\forall (w,x,y,z) \in \mathrm{HPlanes(c)}
$$

Problem Uncertainties and Difficulties

- Organising priorities.
- Solution must be output in reasonable time
- Model exactness
  - Pipe discretisation
  - Compressor polytope





Operation Modes (Example Constraint Set)

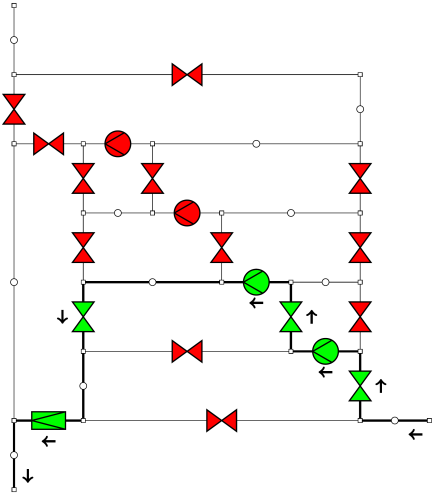
Operation Modes:

- Determines modes and configurations (binary decisions) for all valves and compressors

$M(o, a) := x$ where x is the mode / configuration of arc a in operation mode o $\forall o \in \mathrm{O}$

with $x \in \{\text{open}, \text{closed}\}$ if $a \in \text{Valves}$

$x \in \{\text{bypass}, \text{closed}, \text{cfgs}\}$ if $a \in \text{Compressors}$

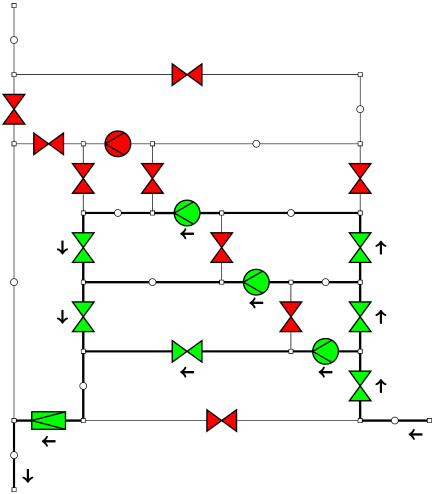


Operation Modes (Example Constraint Set)

Operation Modes:

- Determines modes and configurations (binary decisions) for all valves and compressors
- Limits these mode combinations to a set of size $|\mathrm{O}|$

$$|\mathrm{O}| \quad << \quad 2^{|\text{valves}|} \cdot \prod_{a \in \text{compressors}} (2 + |\text{cfgs}_a|)$$

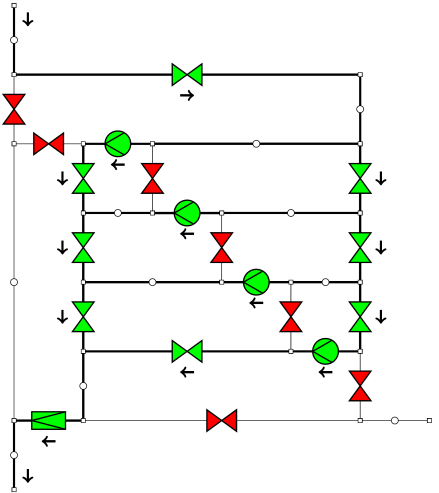


Operation Modes (Example Constraint Set)

Operation Modes:

- Determines modes and configurations (binary decisions) for all valves and compressors
- Limits these mode combinations to a set of size $|\mathrm{O}|$
- Determines the polytope choice for each compressor (Not the active point within)

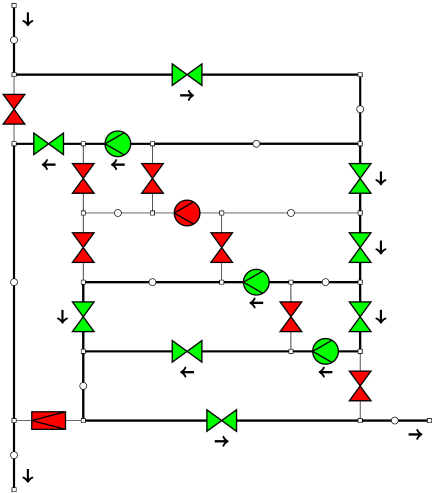
$$m_{c_1} = 0 \quad m_{c_2} = 1 \quad m_{c_3} = 0$$
$$\mathrm{pl}_{c_1} = \mathrm{pl}_{c_3} = \mathrm{pr}_{c_1} = \mathrm{pr}_{c_3} = q_{c_1} = q_{c_3} = 0$$
$$w \cdot \mathrm{pl}_{c_2} + x \cdot \mathrm{pr}_{c_2} + y \cdot q_{c_2} + z \leq 0$$
$$\forall (w, x, y, z) \in \mathrm{HPlanes}(c_2)$$



Operation Modes (Example Constraint Set)

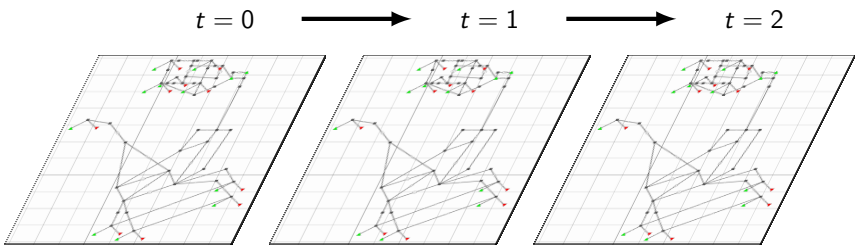
Operation Modes:

- Determines modes and configurations (binary decisions) for all valves and compressors
- Limits these mode combinations to a set of size $|\mathrm{O}|$
- Determines the polytope choice for each compressor (Not the active point within)
- Has an allowed set of flow directions with each choice. For example, the four choices shown never allowed east to north flow.



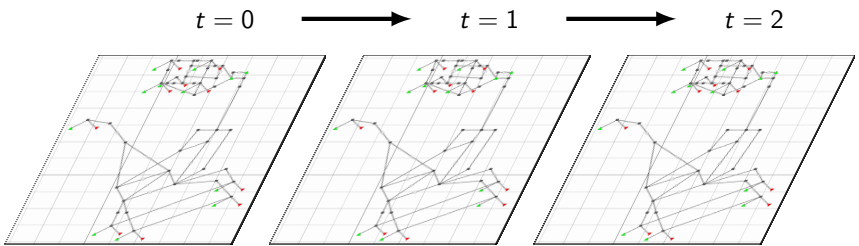
Model Driven Rolling Horizon Approach

- Single large MIP is too unreliable. Solve times can take days to even find a primal solution.



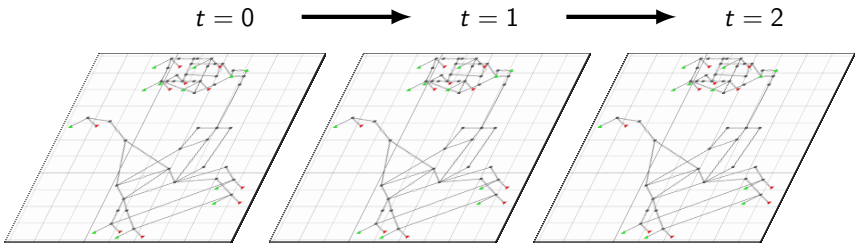
Model Driven Rolling Horizon Approach

- Single large MIP is too unreliable. Solve times can take days to even find a primal solution.
- Idea: Break up by timesteps. Create greedy heuristic for determining best binary decisions



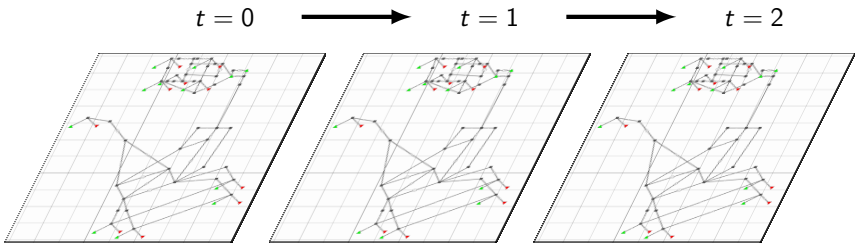
Model Driven Rolling Horizon Approach

- Single large MIP is too unreliable. Solve times can take days to even find a primal solution.
- Idea: Break up by timesteps. Create greedy heuristic for determining best binary decisions
- Introduce a penalty for changing operation modes (set of binary control decisions)



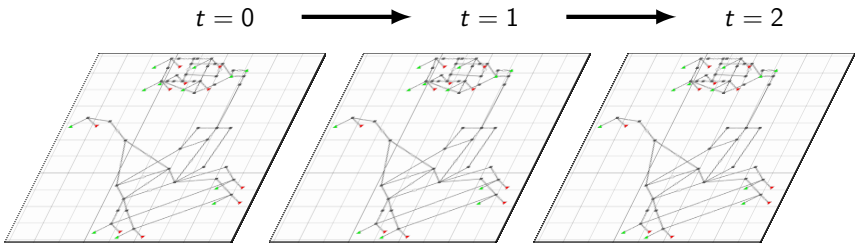
Model Driven Rolling Horizon Approach

- Single large MIP is too unreliable. Solve times can take days to even find a primal solution.
- Idea: Break up by timesteps. Create greedy heuristic for determining best binary decisions
- Introduce a penalty for changing operation modes (set of binary control decisions)
- Create process where series of operation modes can be re-combined if continuity leads to better objective value



Model Driven Rolling Horizon Approach

- Single large MIP is too unreliable. Solve times can take days to even find a primal solution.
- Idea: Break up by timesteps. Create greedy heuristic for determining best binary decisions
- Introduce a penalty for changing operation modes (set of binary control decisions)
- Create process where series of operation modes can be re-combined if continuity leads to better objective value
- Smooth solution and make it feasible w.r.t the larger impractical MIP



Smoothening Procedure

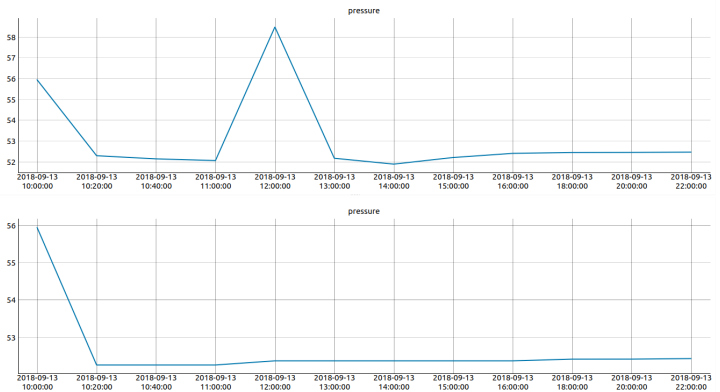
- Downside of greedy approach: Time steps continuous variables are largely independent

Smoothening Procedure

- Downside of greedy approach: Time steps continuous variables are largely independent
- Upside of greedy approach: Quick to solve and end-result scales linearly w.r.t number of time-steps.

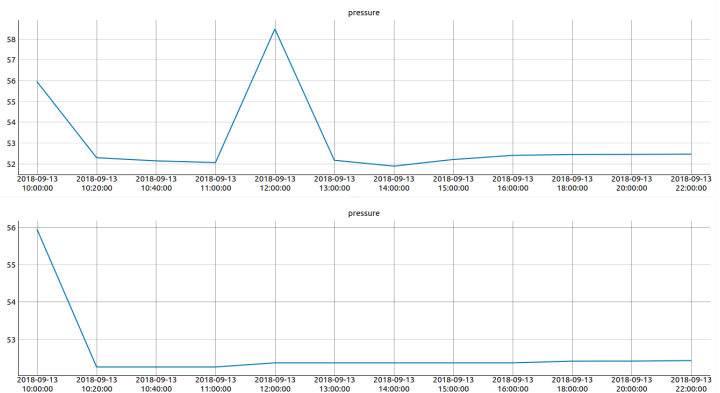
Smoothening Procedure

- Downside of greedy approach: Time steps continuous variables are largely independent
- Upside of greedy approach: Quick to solve and end-result scales linearly w.r.t number of time-steps.
- Solution: Rolling Horizon Approach with fixed binary variables to smooth solution



Smoothening Procedure

- Downside of greedy approach: Time steps continuous variables are largely independent
- Upside of greedy approach: Quick to solve and end-result scales linearly w.r.t number of time-steps.
- Solution: Rolling Horizon Approach with fixed binary variables to smooth solution
- Guarantees primal feasibility to original MIP



Data Driven Dual Neural Network Approach

- Do I have enough data?
  - Substantial amount of data needed for complex tasks
  - Never ending amount of new features and interfaces
  - The first step of a project is often simply creating data readers and ensuring data quality

Data Data Data Data Data Dato Data Data Data
Data Data Data Data Data Data Data Data Data
Data Data Data Data Data Deta Data Data Data
Data Data Data Data Data Data Daqa Data Data
Data Data Data Data Data Data Data Data Data
Data Data Data Gataca Data Data Data Data
Data Data Data Data Data Data Dataa Data Data
Data Data Data Data Data Data Data Data Gato
Data Data Data Data Data Data Data Data Data
Data Data Data Data Data Data Data Data Data
Data Data Data Data Data Data Data Data Data
Data Data Data Dito Data Data Data Data Data
Cata Data acataG Data Data Data Data Data Data
Data Data Data Data Data Data Data Data Data
Dita Data Data Data Data Data Data Data Data
Data Data Data Data Data Data Data Dara Data
Data Data Data Data Data eixt quit kill what exit

Data Driven Dual Neural Network Approach

- Do I have enough data?
- What technique is best for my purpose?
  - More complex techniques aren't always better!
  - There is no universal method (Free Lunch Theorem)
  - Choose method that best fits the problem. Can be from a theoretic approach or preliminary trial and error.

https://xkcd.com/1838/

Data Driven Dual Neural Network Approach

- Do I have enough data?
- What technique is best for my purpose?
- Will a simple heuristic suffice?
  - As seen in this series of lectures, non data-driven heuristics can be very effective
  - Do I need to worry about proven optimality?

https://xkcd.com/1838/

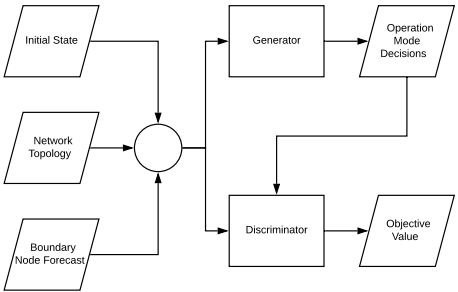
Data Driven Dual Neural Network Approach

- Do I have enough data?
- What technique is best for my purpose?
- Will a simple heuristic suffice?
- How do I model my input?
  - Heavily depends on your method choice
  - Does your data already have intrinsic patterns? E.g. Time repeated values

https://xkcd.com/1838/

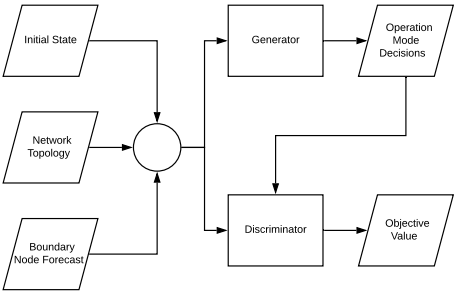
Dual Neural Network Design

- Inspiration
  - Generative Adversarial Networks (GANs) (Learn by having two networks compete against each other)
  - Actor-Critic Methods (Learn with feedback from an environment with an introduced variability in downstream decisions)



Dual Neural Network Design

- Properties of our design:
  - Uses a Generator and Discriminator Design
  - Generator produces binary decisions
  - Disicriminator predicts the induced optimal objective value of a problem with these variables fixed
  - Our MIP/LP solver is the environment
  - Discriminator and Generator are trained at different times, never simultaneously



Pros and Cons of design and implementation

- Pros:
  - Can generate our own data. As we learn our MIP formulation instead of reality itself, data generation is much simpler.
- Cons:
  - Our MIP must accurately represent reality. This may be a problem in extreme scenarios due to simplifications such as our linearisation.

Pros and Cons of design and implementation

- Pros:
  - Can generate our own data. As we learn our MIP formulation instead of reality itself, data generation is much simpler.
  - Learn on a gradient of objective values. This means that our non-optimal decisions are much more likely to be near optimal.
- Cons:
  - Our MIP must accurately represent reality. This may be a problem in extreme scenarios due to simplifications such as our linearisation.
  - Definition of objective function may change. No bound guarantees on solution.

Pros and Cons of design and implementation

- Pros:
  - Can generate our own data. As we learn our MIP formulation instead of reality itself, data generation is much simpler.
  - Learn on a gradient of objective values. This means that our non-optimal decisions are much more likely to be near optimal.
  - Can move a substantial amount of training off-line. Specifically the Discriminator's early training.
- Cons:
  - Our MIP must accurately represent reality. This may be a problem in extreme scenarios due to simplifications such as our linearisation.
  - Definition of objective function may change. No bound guarantees on solution.
  - The Generator requires on-line training to ensure that the Discriminator is returning accurate induced optimal objective values.

Pros and Cons of design and implementation

- Pros:
  - Can generate our own data. As we learn our MIP formulation instead of reality itself, data generation is much simpler.
  - Learn on a gradient of objective values. This means that our non-optimal decisions are much more likely to be near optimal.
  - Can move a substantial amount of training off-line. Specifically the Discriminator's early training.
  - Reduced problem complexity and is used as primal heuristic
- Cons:
  - Our MIP must accurately represent reality. This may be a problem in extreme scenarios due to simplifications such as our linearisation.
  - Definition of objective function may change. No bound guarantees on solution.
  - The Generator requires on-line training to ensure that the Discriminator is returning accurate induced optimal objective values.
  - Output is only binary variables, and the continuous variable values must be found by a MIP/LP solver.

The Last Slide

Thanks for watching!

Combinatorial Optimization @ Work 2020