KTH
VETENSKAP
OCH KONST

Project 2- District Heating Network Optimization

Jagruti Thakur, Mohab Salem, Carolina Ribeiro

Step 1 – Base heating network

- The district heating network will supply heat to **five consumers** using **two heat producers** and **two pipe types**.
  You will use the **Python package DHNx** to model, optimize, and analyze the network.
- Assume all necessary parameters to make the model run correctly, such as:
  - Country of choice- Will be needed in Step 3 based on chosen heat producers.
  - Pipe lengths, investment costs, operating costs, and heat losses.
  - Pipe types available are **DN-30** and **DN-50**.

    Assign losses, capacities, and cost parameters for each pipe type.

    After optimization, ensure that the network **uses both pipe types**.
  - Connection topology
  - Heat demand: **50 heat demand profiles are given in excel**. Select any **five** to represent the consumers’ thermal demand.

Step 1 – Base heating network (continued)

- **Questions to Answer**
  - How does the optimizer decide which pipe type to use in different parts of the network?
  - What is the difference between the selected pipes in terms of **cost, thermal losses, and capacity**?
  - Why the optimizer might favor **one type of pipe** near the beginning of the network or **another type** elsewhere.
- **Learning Objectives**
  - Understand the basic structure and components of a **district heating network**.
  - Learn how to use **DHNx** to model, optimize, and analyze **small-scale heating networks**.

Step 2 – Heating network expansion

- Expand your district heating network to represent a **larger neighborhood**.
- Increase the number of **consumers** from 5 → 10
- Increase the number of **heat producers** from 2 → 5. One of the heat producers should be Heat Pump.
- Use **five pipe types** with different diameters, costs, and heat-loss characteristics:

  DN-25, DN-30, DN-40, DN-50, DN-60
- Update all relevant **input files, parameters, and assumptions**.

  Ensure that the expanded model is **technically feasible** and can be optimized successfully.

Step 2 – Heating network expansion (continued)

- **Questions**
  - How did increasing the number of consumers and producers affect the **network topology** and **pipe selection**?
  - Did the optimizer choose different **pipe types** compared to Step 1? Why?
  - Are the additional **producers** utilized efficiently, or are some underused?
  - How are the **total investment costs**, **operating costs**, and **losses** compared with the smaller network in Step 1?
- **Learning Objectives**
  - Learn how to **expand** a district-heating model to represent a larger system with more consumers and producers.
  - Assess how optimization behavior changes with additional **pipe types** and **heat producers**.
  - Understand how scaling affects **network topology**, **energy distribution**, and **computational complexity**.

Step 3 – Energy system optimization (Oemof)

- Using the network developed in **Step 2**, model the **five heat producers** in **oemof.solph** to meet the **aggregate heat demand**.
- Calculate aggregated load based on chosen demand profiles in step 2
- Specify the **technologies** for each head producer.
- Add technical and economic parameters for the technologies (One possible source for data- https://ens.dk/en/analyses-and-statistics/technology-data-generation-electricity-and-district-heating)
- Define **capacities** for each generation unit with **realistic limits**, ensuring the system can satisfy the total load.
- For a **solar collector**, create your own **solar irradiance profile (Choose your own country)**.
- **Do not include any type of thermal storage** in this step.

Step 3 – Energy system optimization (Continued)

- **Questions**
  - Which generating units are supplying the majority of the heat demand over the simulation period? How is the load distributed among the units?
  - Are there periods when any of the generating units reach their maximum defined capacity? If yes, which units, and during what times?
  - What are the total operational costs for each unit, and which unit contributes most to system costs?
- **Learning Objectives**
  - Learn how to model and simulate multi-technology heat generation systems using oemof.solph.
  - Understand interactions between different generation units (boilers, heat pumps, etc.) in meeting variable demand.
  - Analyze model outputs to evaluate system performance
  - Develop the ability to interpret operational patterns, capacity use, and efficiency trends.

Step 4 – Integration of district heating network (DHNx) with the *energy system model* (oemof)

- Combine your **DHNx** and **oemof.solph** models so that the generation system accounts for **distribution losses** from the district heating network.

This step connects the *physical network model* (DHNx) with the *energy system optimization model* (oemof).

- **How to Set Up the Link**
  - **Step 2** output file contains loss data for each pipe or the total network.
  - Export or record the **total thermal losses** (in MWh or kWh).
  - **Prepare the aggregated demand**
    - In Step 3, your oemof model used the **total consumer demand** (from the ten consumers).
    - Now **add the distribution losses** to this demand profile: $Q_{total} = Q_{demand} + Q_{losses}$
    - Create a new input file or column for the **adjusted load profile** that includes both aggregated heat demand + losses.

Step 4 – Link setup continued

- **Update the oemof model inputs**
  - Replace your previous demand profile in oemof with the new adjusted total load.
  - Keep the same five generating units (from Step 2 and 3).
  - You don’t need to modify the network topology.
- **Run the oemof simulation**
  - Execute your oemof model again using the updated demand.
  - Compare results such as total heat production, generator dispatch, and system costs before and after including losses.
- **Questions**
  - How does including distribution losses change the operation of each generating unit?
  - Do any units now operate closer to their capacity limits?
  - How are system efficiency, operating cost, and total heat generation affected?

Step 4 – Link setup continued

- **Learning objectives**
  - Learn how to integrate distribution losses from the district heating network in the energy systems optimization model.
  - Understand how distribution losses impact generation dispatch.
  - Evaluate effects on total energy use, capacity needs, and system efficiency.
  - Build skills in linking network physics with system optimization models

Step 5 – Environmental Optimization

- Extend your **oemof** model to include **CO_2 emission costs** for each generating unit, and explore how this affects the dispatch of technologies.

  You will run **two scenarios**:

  - **Cost + Emission Costs** → both fuel and emission costs are considered.
  - **Emission Costs Only** → all fuel costs set to 0 to isolate the effect of emissions.

- **Keep the energy system model from Step 4 and:**
  - Use the same network setup as in Step 4 and the **adjusted thermal demand** (including DHNx losses).
  - Keep all five head producers and their capacity definitions.
  - The only change now is to add **emission costs**. For all the technologies.

Step 5 – Environmental Optimization (continued)

SCENARIOS

Scenario A – Total Cost + Emission Costs

- Keep your **original investment and operating costs** from Step 4.
- Add emission costs to each unit's variable cost
- Run the model normally (no change to the objective function).
  oemof will still **minimize total system cost**, which now includes emission costs.

- **Scenario B – Emission Costs Only**
  - Set all **fuel/operating costs = 0**.
  - Keep **emission costs** as defined above.
  - Run the same model again.
  - This isolates the **environmental driver**: dispatch decisions now depend *only* on CO_2 emissions.

Step 5 – Environmental Optimization (continued)

- **Questions**

  After running both scenarios, compare:
  - In Scenario A, which units become less competitive when emission costs are included?
  - In Scenario B, which low-carbon units dominate operation?
  - What is the trade-off between economic and environmental outcomes considering Step 4 and Step 5 (both scenarios)?
- **Learning Objectives**
  - Learn how to integrate **CO_2-related costs** into energy system optimization analysis, using oemof, without changing the optimization objective.
  - Understand how emission pricing affects **dispatch decisions** and **system composition**.
  - Compare purely **economic vs. environmental** optimization behavior.
  - Build intuition for **carbon pricing** as a policy and modeling concept.

Tools for Project 2

- **oemof**: A general-purpose energy system modeling and optimization framework
- **DHNx**: A specialized extension for district heating and cooling network optimization
- Both tools are open-source, Python-based, and designed to work together for comprehensive energy system analysis.

KTH VETENSKAP OCH KONST

oemof (Open Energy Modelling Framework)

oemof is a Python toolbox for energy system modeling and optimization that uses a modular, open-source design for cross-sectoral energy system analyses.

It provides a flexible framework to model complex energy systems across multiple sectors (electricity, heat, gas, etc.).

Core Concepts

- **1. Graph-Based Structure**
  - Energy systems in oemof are represented as graphs with buses (representing energy carriers) and components (representing technologies) connected by flows.
    - **Nodes**: The basic building blocks (buses and components)
    - **Edges**: Directed connections representing energy flows
    - **Flows**: Values attached to edges representing energy quantities

oemof ['ø:mɔf]
open energy modelling framework

A modular open source framework to model energy supply systems

https://github.com/oemof/oemof

oemof (Open Energy Modelling Framework) (cont)

- **2. Three main elements of the energy system model:** Buses, components and flows.

  Buses

  - Buses are balance objects where every component must be connected, and the sum of all inputs must equal the sum of all outputs within each time step.
  - Think of buses as:
  - Energy carriers (electricity, heat, gas, hydrogen)
  - Connection points where supply and demand meet
  - Balance nodes ensuring energy conservation

python

```python
import oemof.solph as solph

# Create buses for different energy carriers
b_electricity = solph.Bus(label="electricity")
b_heat = solph.Bus(label="heat_35C")
b_gas = solph.Bus(label="natural_gas")
```

oemof (Open Energy Modelling Framework) (cont)

- **2. Main elements of the energy system model (cont)**

  Components

  - Components are technologies that transform, generate, consume, or store energy:
  - Sources: Energy producers (solar PV, wind turbines, grid connection)
  - Sinks: Energy consumers (demands, exports)
  - Converters: Technologies that transform energy (CHP, heat pumps, boilers)
  - Storages: Energy storage systems (batteries, thermal storage)

  Flows

  - Flows connect components to buses and can have various constraints like costs, capacities, and temporal patterns.

python

```python
# Example: Heat pump as a converter
heat_pump = solph.components.Converter(
    label="heat_pump",
    inputs={b_electricity: solph.Flow()},
    outputs={b_heat: solph.Flow()},
    conversion_factors={b_heat: 3.5}  # COP = 3.5
)
```

```python
# Flow with constraints
electricity_demand = solph.components.Sink(
    label="electricity_demand",
    inputs={b_electricity: solph.Flow(
        nominal_value=100,  # kW
        fix=[0.5, 0.6, 0.7, 0.8],  # Load profile
        variable_costs=0  # €/kWh
    )}
)
```

oemof.solph: The Optimization Module

oemof-solph facilitates the formulation of (mixed-integer) linear programs for energy system optimization, allowing multi-sector models with user-defined time resolution.

- **What it does:**
  - Converts your energy system graph into mathematical optimization problems
  - Solves for least-cost operation or investment decisions
  - Handles unit commitment, dispatch optimization, and capacity planning
- **Key Features of oemof**
  - **Multi-sector modeling**: Electricity, heat, gas, transport in one model
  - **Flexible time resolution**: Hours, days, representative periods
  - **Open source**: MIT license, transparent, collaborative
  - **Modular design**: Extensible with custom components
  - **Mathematical foundation**: Linear programming (LP) and mixed-integer linear programming (MILP)

OEMOF Answers:

- Which technologies should operate at what time?

Should electricity come from solar, wind, or gas at 2 PM?
Should heat be provided from CHP or Geothermal plant at 1 PM?

- How much energy should flow between components?

How much heat should go from the boiler to the storage tank or to the district heating network?

- What is the optimal system configuration?
- *Should we invest in a heat pump, more storage, or solar collectors?*

- What are the total costs and emissions of the system?

- How does the system behave under different scenarios?

What if fuel prices rise? or What if we increase renewable energy targets?

oemof.solph- Recap

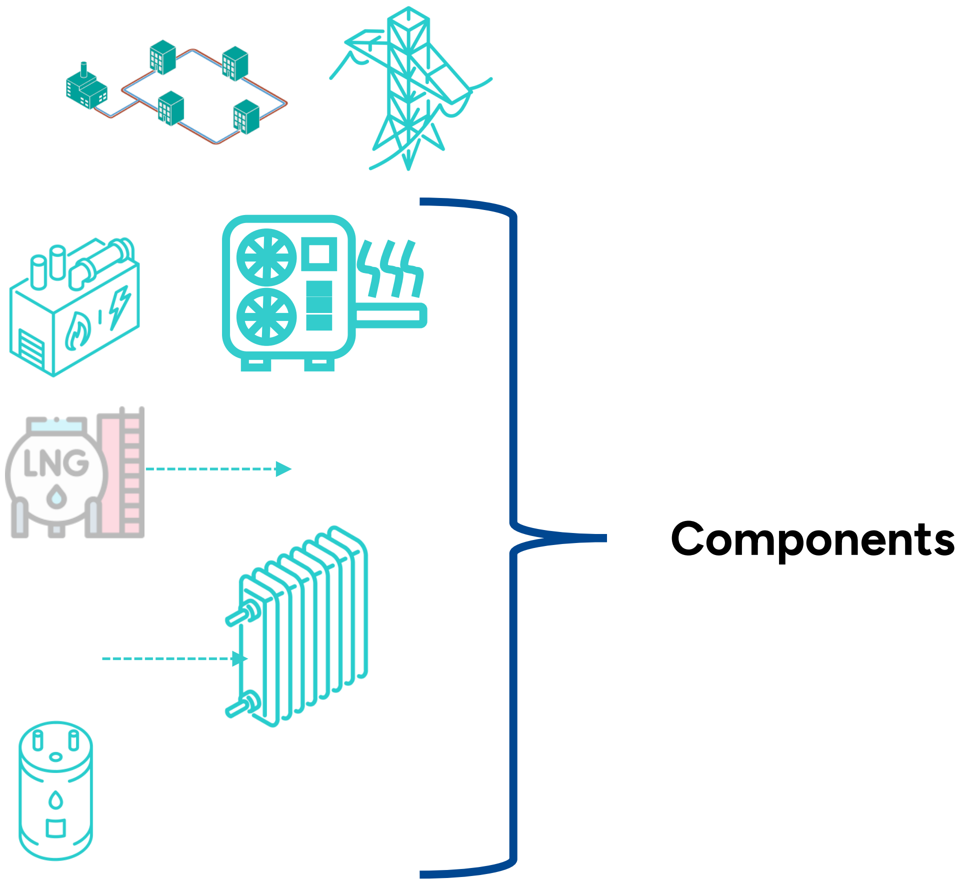
Buses : *represent commodities such as electricity, heat or natural gas. The main aim of buses to balance input and output flow at any given point in time.*

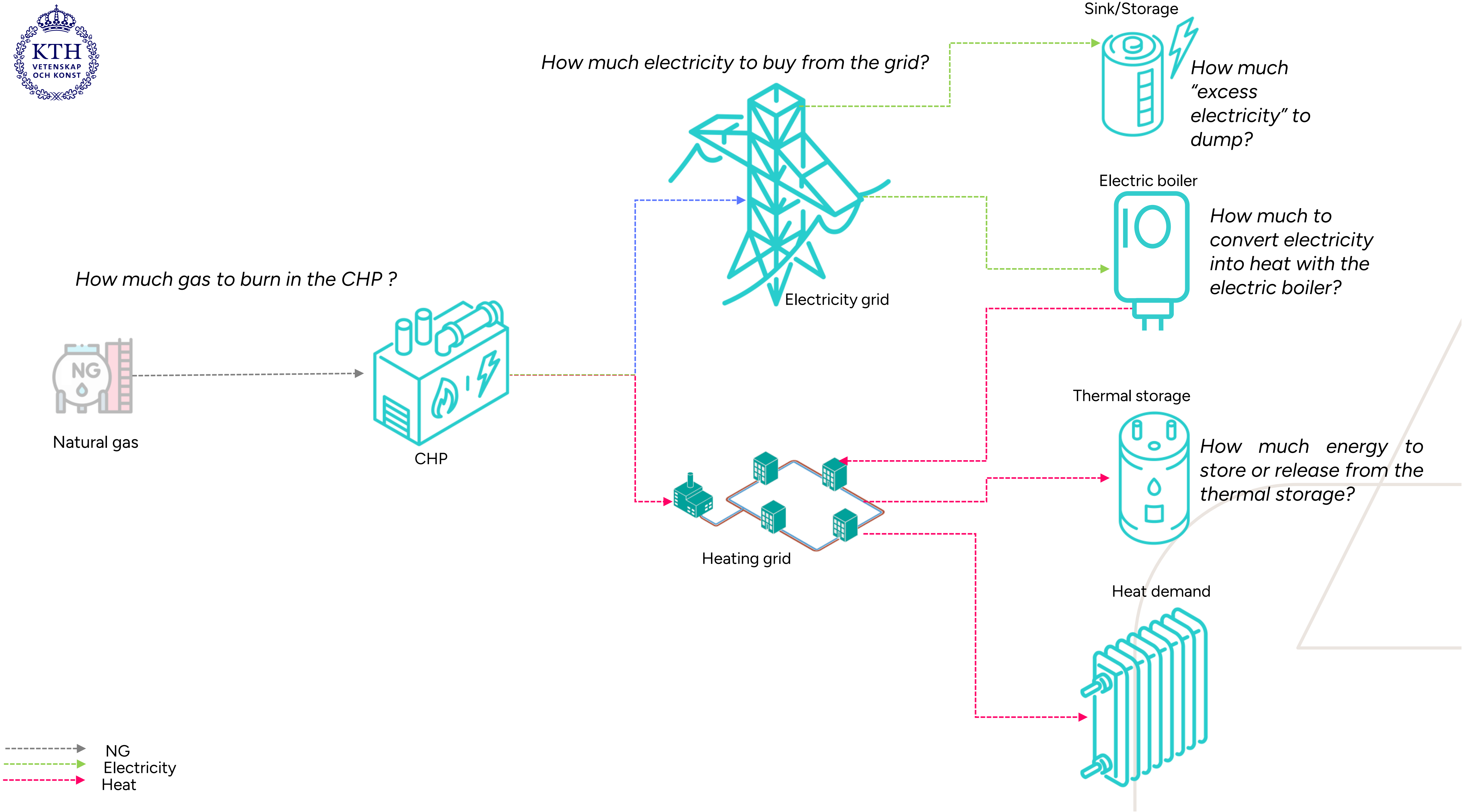
Converters : *are like heat pumps, combined heat and power plants or electrolyzers.*

Sources : *Does not require any inputs, for example: PV panels or electricity imports from the grid.*

Sinks : *Does not generating any output, for example: Heat or electricity demand of consumers.*

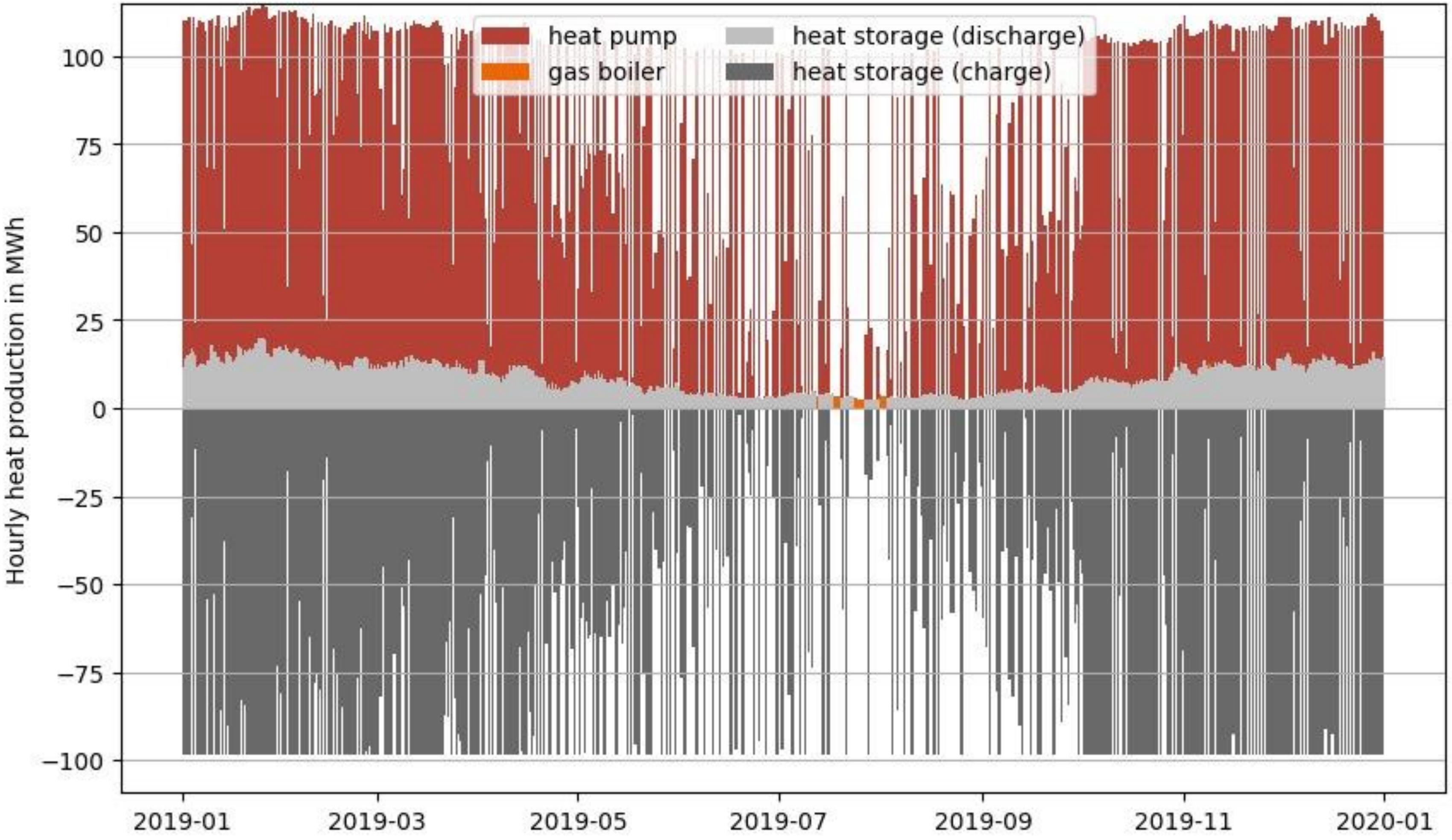
Energy storage : *Such as batteries or heat storage tanks.*





Sink/Storage
How much electricity to buy from the grid?
How much "excess electricity" to dump?
Electric boiler
How much to convert electricity into heat with the electric boiler?
Electricity grid
How much gas to burn in the CHP ?
NG
Natural gas
CHP
Thermal storage
How much energy to store or release from the thermal storage?
Heating grid
Heat demand
NG
Electricity
Heat

Example- Results



DHNx

- **What is DHNx?**
  - DHNx is an open toolbox specifically designed for district heating and cooling network optimization and simulation, providing models for both investment planning and operational analysis.

Challenge of Heat Networks

- District heating networks have unique characteristics that general energy models don't capture well:
  - **Spatial structure**: Physical pipe networks with specific layouts
  - **Heat losses**: Temperature-dependent losses along pipes
  - **Pressure losses**: Hydraulic constraints affecting pumping costs
  - **Network topology**: Tree structures, routing decisions
  - **Pipe sizing**: Investment trade-offs between diameter and heat loss

oemof/**DHNx**

District heating system optimisation and simulation models

https://dhnx.readthedocs.io/en/latest/

DHNx adds district heating pipes to existing energy systems, where each pipe is modeled as a transformer with energy input and output accounting for heat losses.

DHNx - Core Components

1. **ThermalNetwork Class -** The central object that represents your district heating system:

- **Components of a ThermalNetwork:**
  - **Pipes**: Connect nodes, have length, diameter, heat losses
  - **Producers**: Heat generation points (plants, substations)
  - **Consumers**: Heat demand points (buildings)
  - **Forks**: Junction points where pipes meet

2. Input Data Structure

- DHNx requires two groups of data:
  - network components and connectivity (for ThermalNetwork class), and
  - investment/operational data for optimization.

python

```python
import dhnx

# Create thermal network from data files
thermal_network = dhnx.network.ThermalNetwork()
thermal_network.from_csv_folder('path/to/network_data')
```

Required CSV files:

```
project/
├── consumers.csv       # Building demands, locations
├── producers.csv       # Heat sources, capacities
├── pipes.csv          # Pipe connections, lengths
├── forks.csv          # Network junctions
└── sequences/
    ├── consumers-heat_demand.csv
    └── environment-temp_env.csv
```

DHNx – Two main models

A. Optimization Models

For network design and investment planning, DHNx provides optimization setup that returns oemof components to be added t existing energy systems.

- **What you can optimize:**
  - Network topology (which pipes to build)
  - Pipe diameters (investment vs. operating costs)
  - Producer capacities
  - Connection decisions (which buildings to connect)

python

```python
# Load investment options
invest_opt = dhnx.input_output.load_invest_options('path/to/invest_options

# Run optimization
thermal_network.optimize_investment(invest_options=invest_opt)

# Access results
results = thermal_network.results.optimization['components']['pipes']
```

B. Simulation Models

For detailed physical analysis, simulation models address question: about heat losses depending on temperatures, pressure losses, an pump energy requirements.

- **What you can simulate:**
  - Heat losses along the network
  - Temperature distribution
  - Pressure losses
  - Pumping power requirements

python

```python
# Run simulation
thermal_network.simulate()

# Results include temperature, heat loss, pressure for each pipe
```

DHNx - Features

Heat Losses

- Heat losses are calculated based on inlet/outlet temperatures, environmental temperature, and thermal transmittance, using steady-state assumptions.
- Temperature drop equation:
  - T_out = T_env + (T_in - T_env) * exp(-U*A / (m_dot * cp))

Pipe Sizing

  - Multiple diameter options with different:
  - Investment costs (€/m)
  - Heat loss coefficients (W/m/K)
  - Pressure loss characteristics

- **Integration with oemof**
  - DHNx uses oemof-solph optimization and results can be accessed through oemof's labeling system, with automatic tagging by spatial belonging (consumers, producers, infrastructure).

DHNx - Example

Investigate how to design a cost-efficient district heating network that connects multiple heat producers to various heat consumers.

The goal is to determine which pipelines should be built and at what capacities to minimize the total system cost, while ensuring all consumer heat demands are satisfied.

You must determine the optimal network configuration by solving an investment optimization problem that minimizes the total system cost while satisfying all technical and physical constraints.

Input data

The input data is provided in two folders:

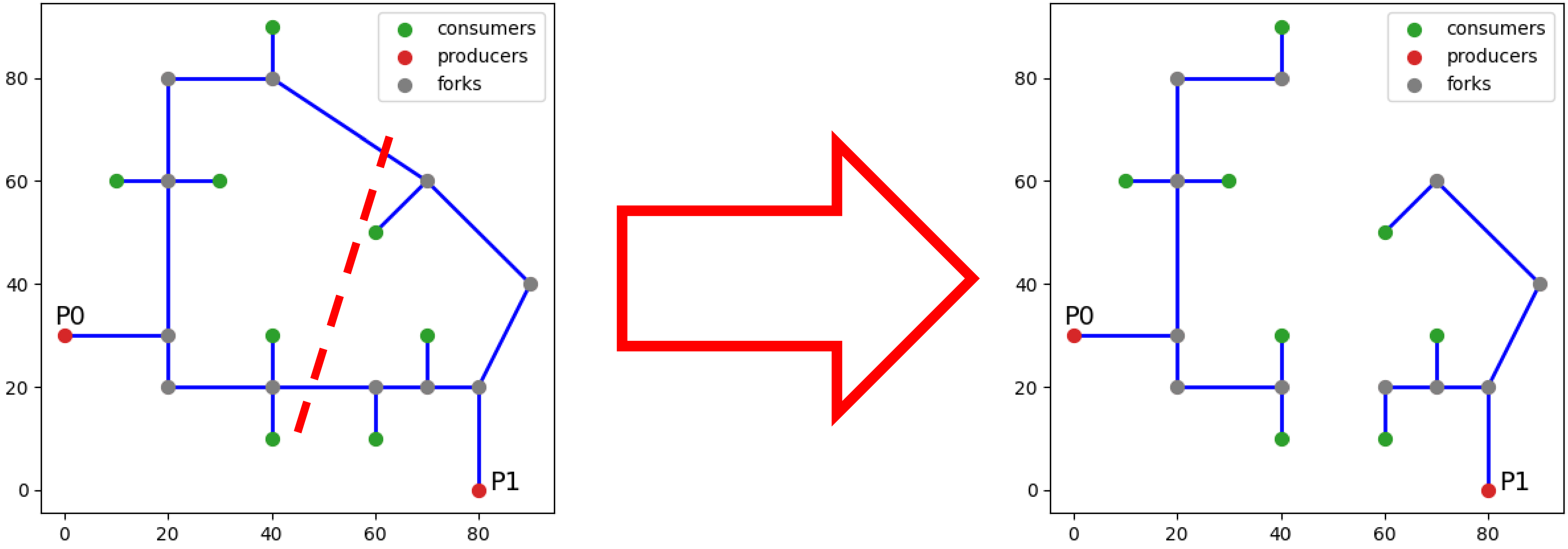
- *twn_data* includes:
  - *Locations (longitude, latitude) of producers, consumers, and forks.
  - *Possible connections (edges) between nodes.
  - *Technical parameters for each possible connection (length, losses, and maximum capacity).

- *invest_data* includes:
  - *Cost of installing different pipe types (euros/m).
  - *Operational costs.
  - *Thermal loss factors.

Questions:

- What is the optimal configuration of the district heating network (which pipes are selected)?
- What are the capacities, directions, and thermal losses for each active pipe?
- What is the minimum total system cost (objective value) obtained from the optimization?
- How does the optimized network layout differ from the initial full network?

DHNx - Example



DHNx - Recap

- Network design / investment: *Which pipes should be built? What diameters (sizes) should they have?*

- Cost optimization: *What is the least-cost configuration (investment + operation) of the network?*

- Thermal loss analysis: *How much heat is lost along the pipes, and how does this affect efficiency and cost?*

- Flow optimization: *How much heat should flow through each pipe to satisfy all consumer demands efficiently?*

- Scenario testing: *What happens to the system cost and performance if demand increases or supply temperature decreases?*

Thank you