Energy System Optimization

Linear Optimization for Energy System Planning and Analysis

Jonathan Chambers

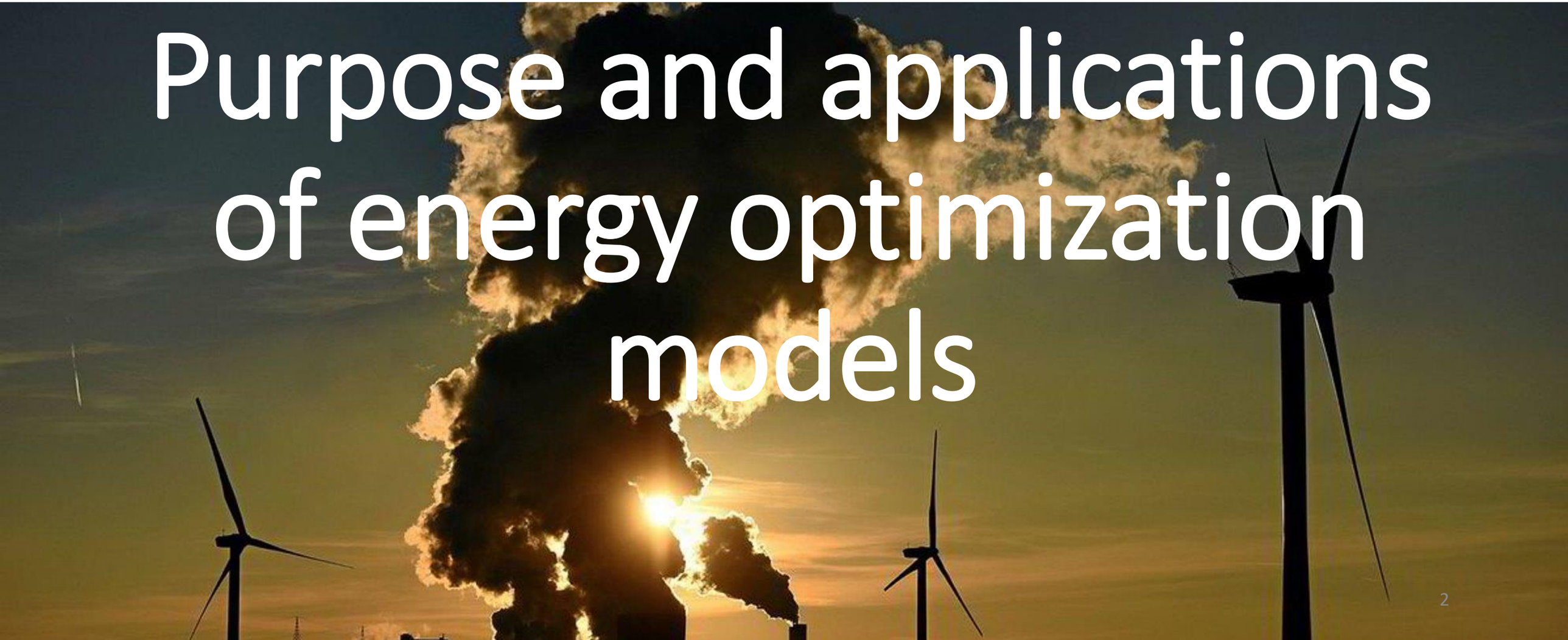
Methods for analyzing energy efficiency and renewable technologies

Learning goals

- Understand **why and when linear optimization can be useful** for energy system modeling
- Be able to describe the **structure** of an optimization model (sets, parameters, variables, constraints, objective)
- Be able to **formulate scalable optimization models**
- Gain intuition on how the price on CO2 emissions has the potential to impact the cost-effectiveness of the energy transition (assignment)

Part I:

Purpose and applications of energy optimization models



Optimization

- Maximize or minimize an objective function
  - Cost, emissions, welfare, etc
- By identifying the optimal values of the variables that the objective function depends on
  - Power plant operation, new installed capacity, power flows, etc
- Subject to certain system constraints
  - Supply equals demand, cannot produce 2GW with a 1GW power plant, etc

Purpose and applications

- Linear optimization models are useful to model a broad variety of energy systems
- The optimization allows to approximate perfect (electricity) markets
- The design of the model is determined by both the system we want to represent and the questions we want to answer

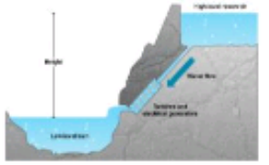
Input and Output of an Energy Systems Model

Scenario assumptions / energy economic market view

- Power plant technology development (convention-al expansion candidates: capex, opex, efficiency...)
- Technology development data for Renewables (capex, opex)
- (Regional) renewable potentials and generation profiles
- World market fuel price scenarios (e.g. hard coal, gas, oil)
- Domestic fuel price scenarios (e.g. lignite)
- CO_2 certificate limits or prices
- Macroeconomic developments (GDP, inflation, exchange rates...)
- (Regional) electricity consumption (amount and profile)

Existing assets and decided projects

Thermal power plant park (e.g. Germany, Europe: Inst. Capacities, Efficiencies...)

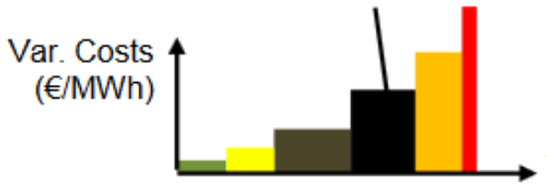


Hydro storages (Volumes, discharge rates)

Market Model

Minimisation of system expenses for load coverage

System (residual) load –



External (policy-driven) developments (e.g. nuclear phase-out in Germany)

Results

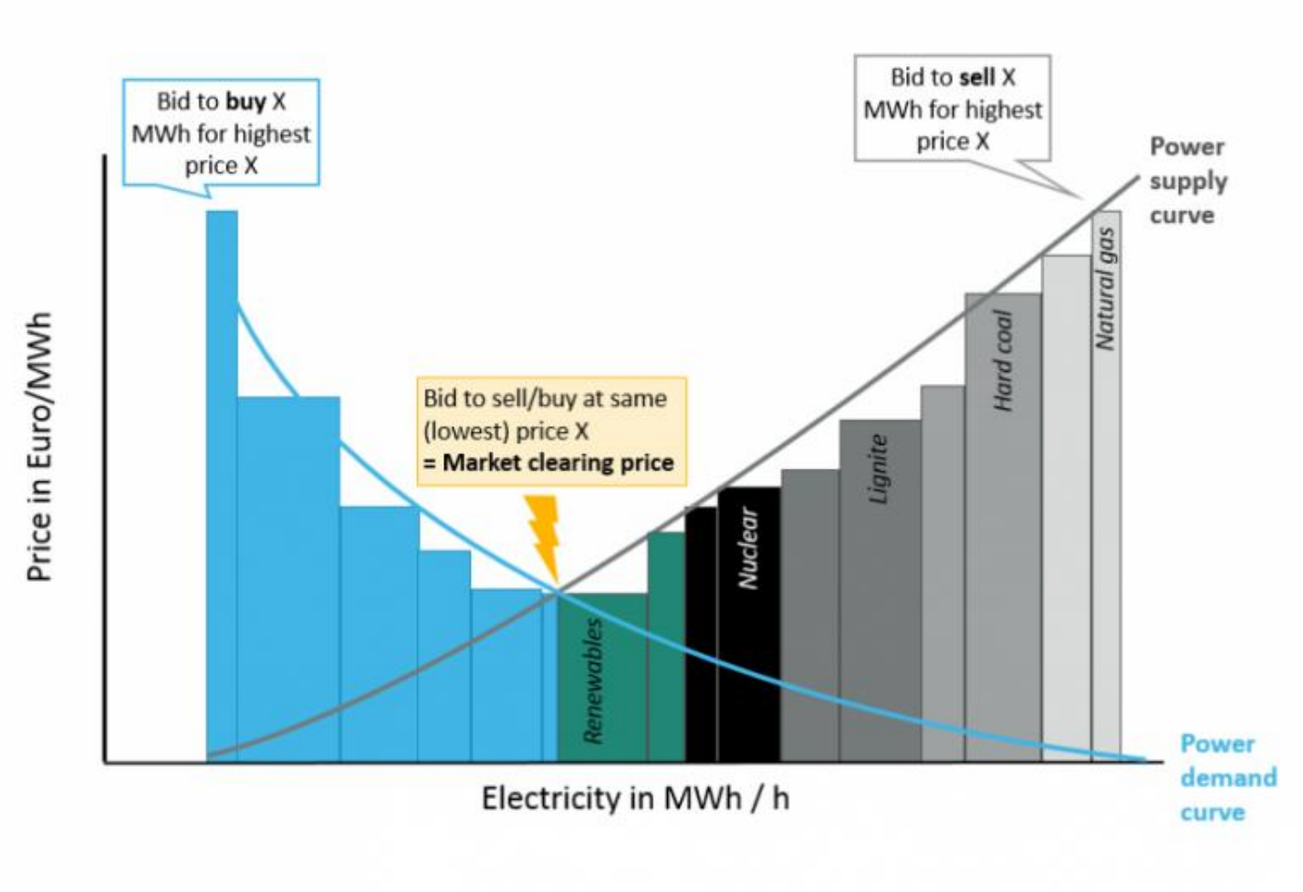
- (Regional) power plant park development
- Fuel consumption
- CO_2 emissions
- Marginal prices (short-run and long-run marginal costs)
- Total system expenses

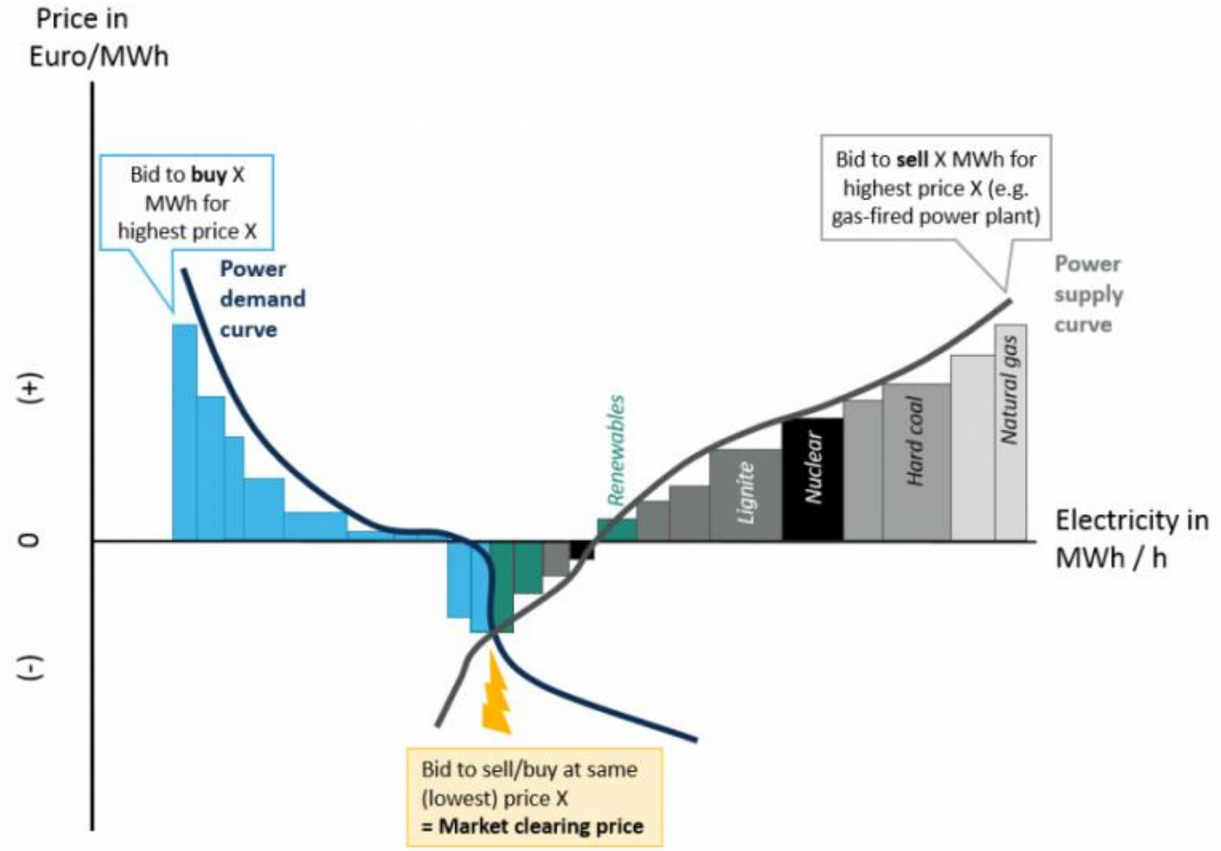
[Bertsch, 2015]

Market theory

- Markets minimize costs to cover demand (perfect markets)
- Optimization can be used to approximate market behavior
- Answer questions like:
  - How does the market react if we change certain parameters?
  - Estimate future electricity prices
  - Analyze drivers
  - Controlled parameter changes to understand market behavior

Market price set by supply and demand



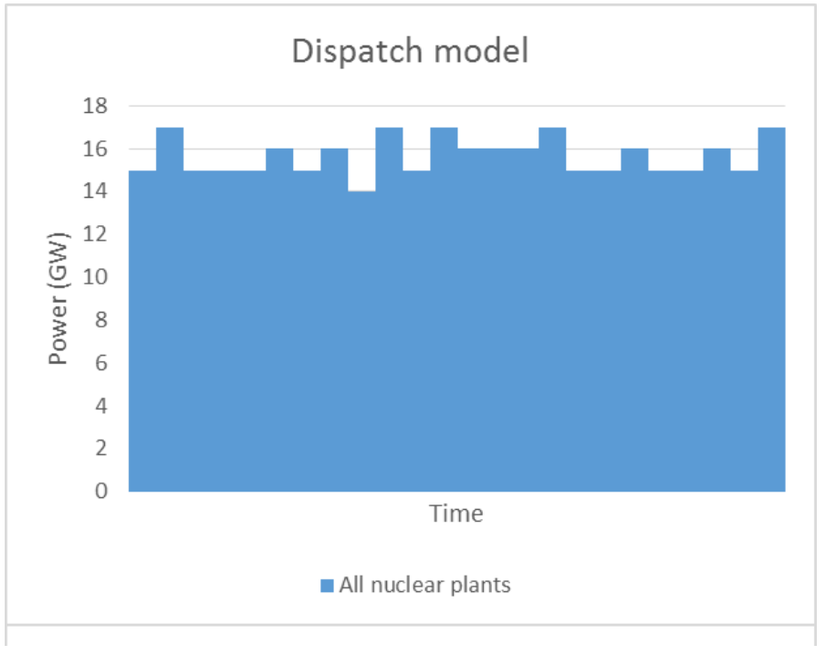


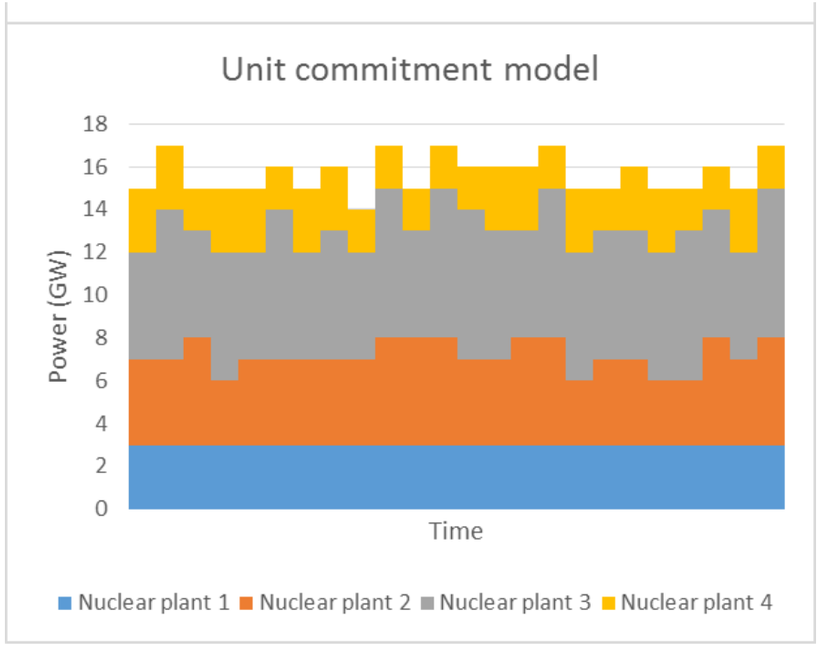
All models are wrong, but some are useful!

- Known limits to market optimization method, but still very useful!
- We can explore relative changes, sensitivity to study **trends and counterfactuals.**
- This is different from **prediction.**
- Answer questions like:
  - How does the market react if we change certain parameters?
  - Estimate future electricity prices
  - Analyze drivers
  - Controlled parameter changes to understand market behavior

Some types of models

- Dispatch vs. unit commitment
  - *Assignment: typical simple dispatch model*



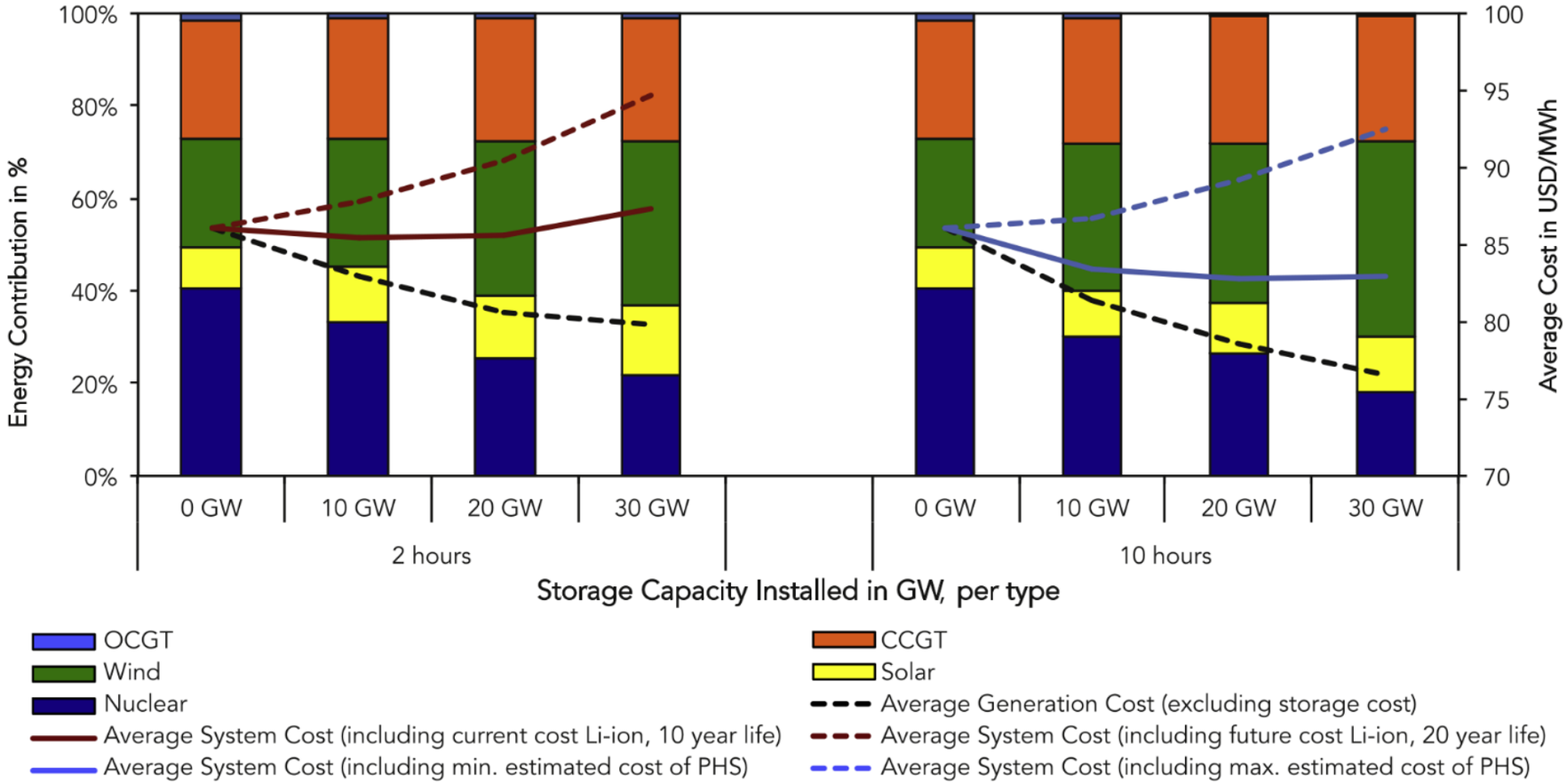


Some types of models

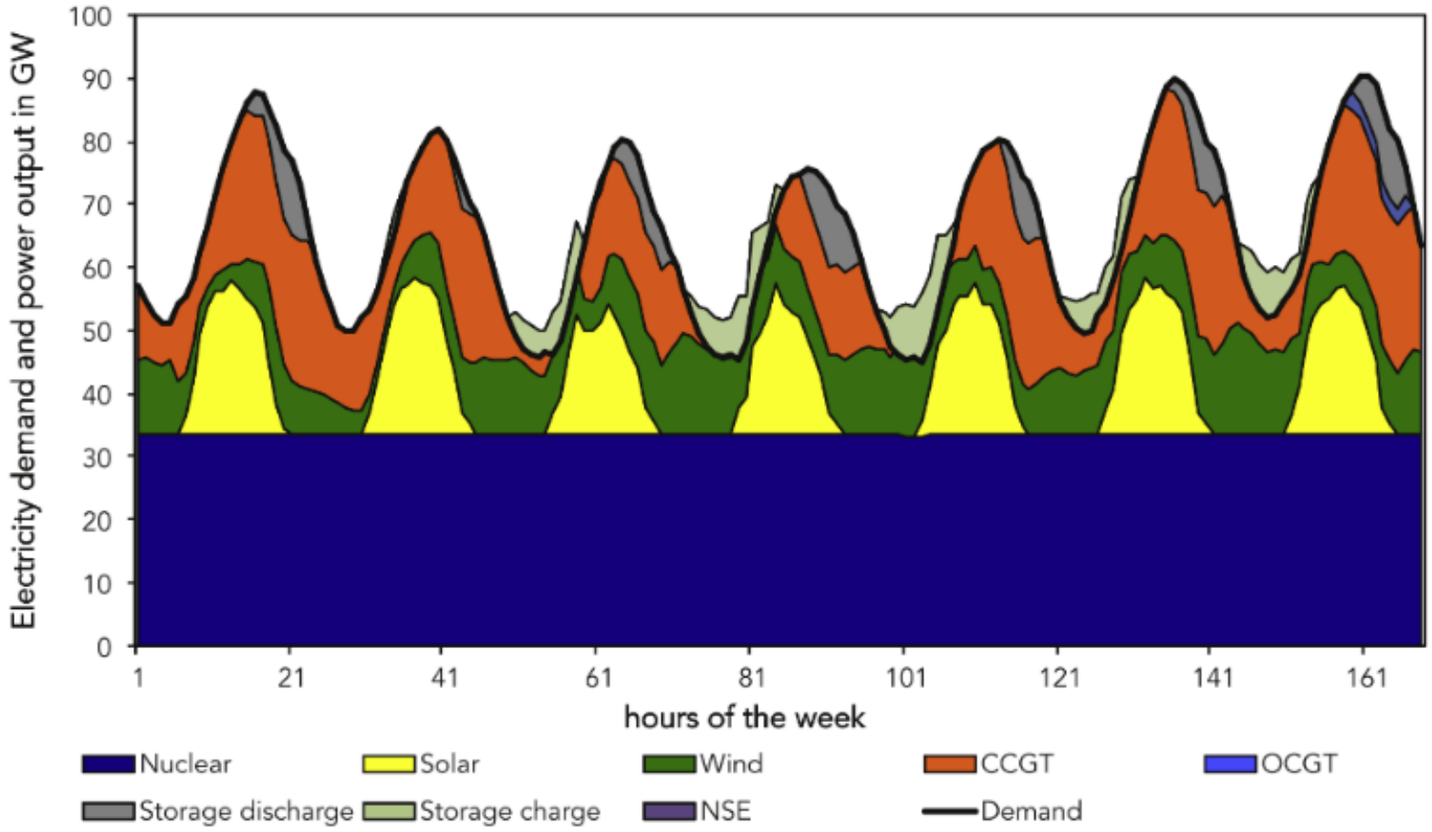
- Temporal resolution: aggregated year, hourly, representative weeks/days, etc
  - *Assignment: full year, season*
- Energy system operation only vs. capacity expansion/retirement
  - *Assignment: integrated operation/capacity investment*
- Greenfield (optimize full system configuration) vs. brownfield (optimize capacity starting from existing system)
  - *Assignment: brownfield with scenarios*

Examples from research: adding grid level storage in Texas

- Approximate the year through 4 representative weeks
- Assume no pre-existing storage
- How does the optimal system composition and costs change if storage is added?

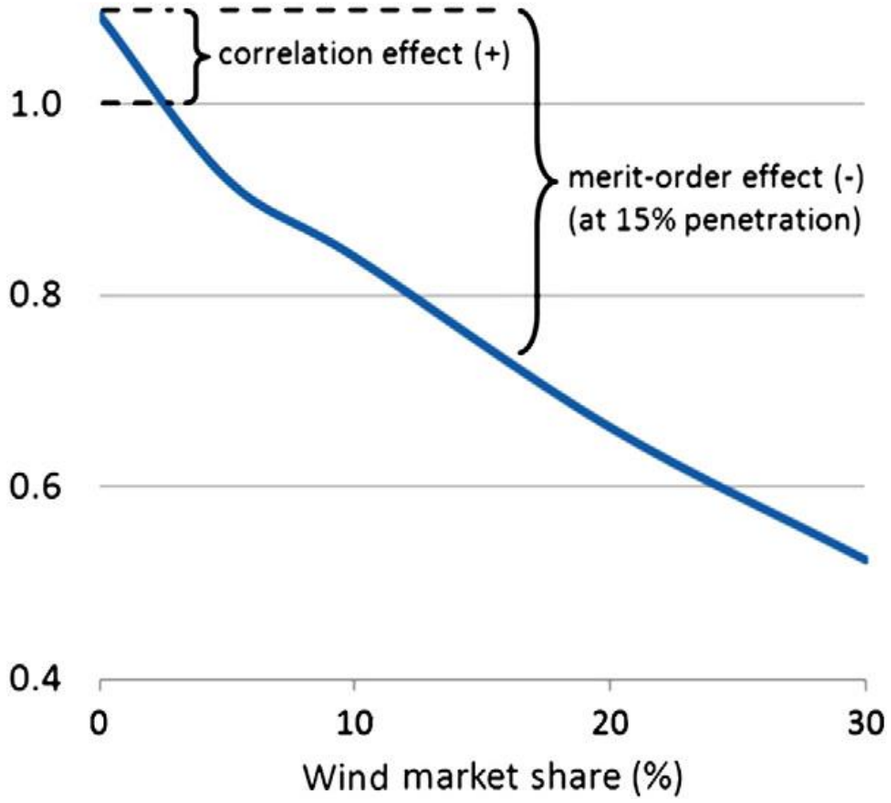


Examples from research: adding grid level storage in Texas



Examples from research: Market value of variable renewables

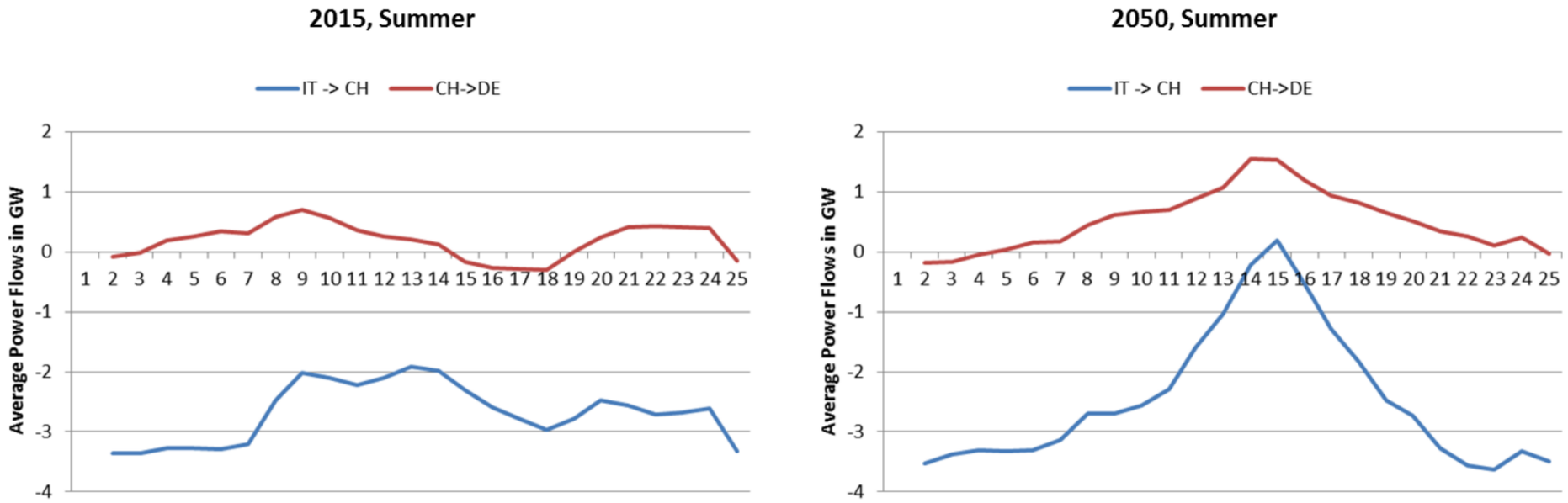
- Dispatch model minimizing cost for each hour of a single year; multiple western European countries
- How does the value/revenue of wind and solar change if the capacities are increased?
  - At 30% penetration, electricity from wind is worth half of that from a constant source of electricity.



Examples from research: Future role of Switzerland

- Detailed Swiss market model, all hydro power plants

Figure 4: Solar power as future driving factor of Swiss border balances in Summer



Part II:

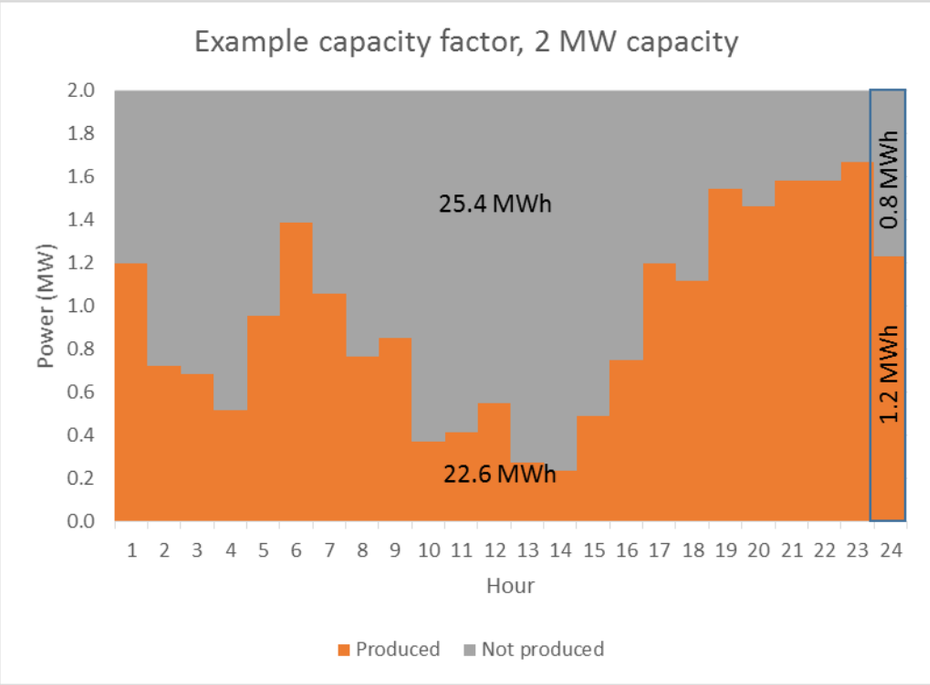
Important concepts

Fixed costs, variable costs

- Fixed Costs (FC)
  - Constant costs over time
  - Include annualised CAPEX costs
- Variable Costs (VC)
  - Variable in time
  - For example price of gas for power plant

Capacity factor

- Ratio of produced energy and maximum energy produced at full nominal capacity
- What causes electricity generators to have a capacity factor < 1?



Example: 2 MW capacity producing at varying output for 24 hours

- *Daily CF:*

$$\frac{22.6\text{MWh}}{25.4\text{MWh} + 22.6\text{MWh}} = 47.2\%$$

- *Hourly CF (hour 24):*

$$\frac{1.2\text{MWh}}{0.8\text{MWh} + 1.2\text{MWh}} = 60\%$$

Full load hours (FLH)

- FLH = Capacity factor * total duration
- How long would we need to operate at full power to produce the respective amount of energy?
- Units: hours or (produced MWh)/(MW installed capacity)

Example Daily FLH:

$$47.2\% \cdot 24\ \mathrm{h/day} = 11.3\ \mathrm{hours/day}$$

The marginal generator

- As energy demand varies, different energy generators are activated or disactivated.
- We assume the generators are activated in order from the least expensive to the most expensive
- This is called the **merit order**
- The **marginal generator is** the **most expensive** generator that needs to come online to meet the demand
- It sets the price for **everyone**

The merit order effect

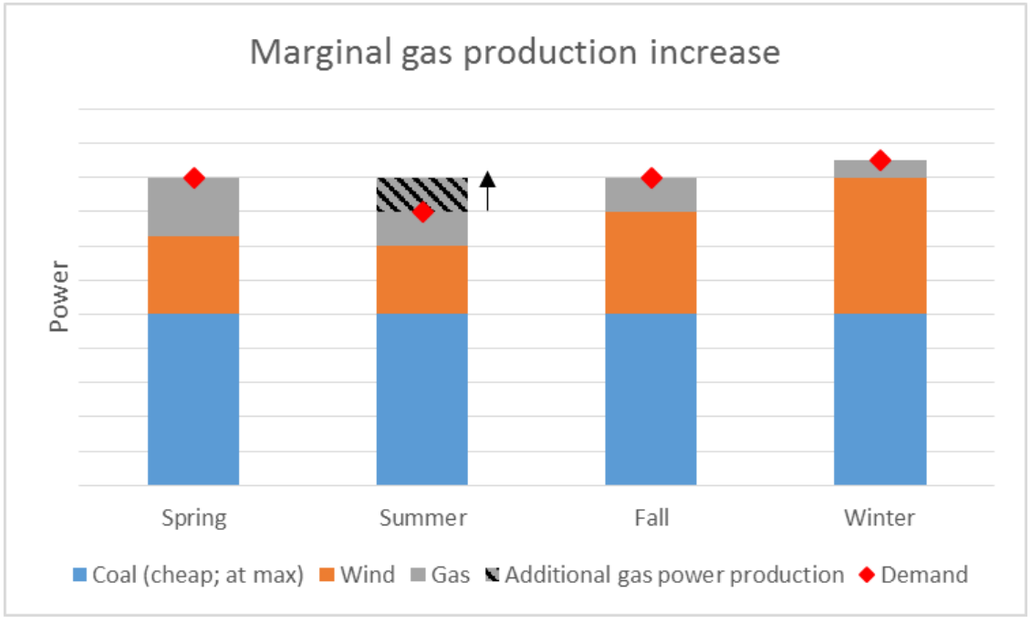
- The sorted variable costs of power plants form the **supply cost curve**
- **Short term variable costs** of power plants determine the power price for a certain demand
- The **marginal generator** sets the price
  - Example: wind and solar power with low short term variable costs reduce electricity selling prices

Shadow prices

- “Shadow price” refers to the calculated price of something that is not directly traded in a real marketplace.
- In the energy system, the ‘ideal’ or ‘optimal’ price of electricity at a given time is the price of the **marginal generator**
  - How much does the electricity price (CHF/MW) change if the demand is increased
- In our energy system model, we set the market electricity price equal to the shadow price
- The real energy prices may be different
  - e.g. on the European Energy Exchange EEX

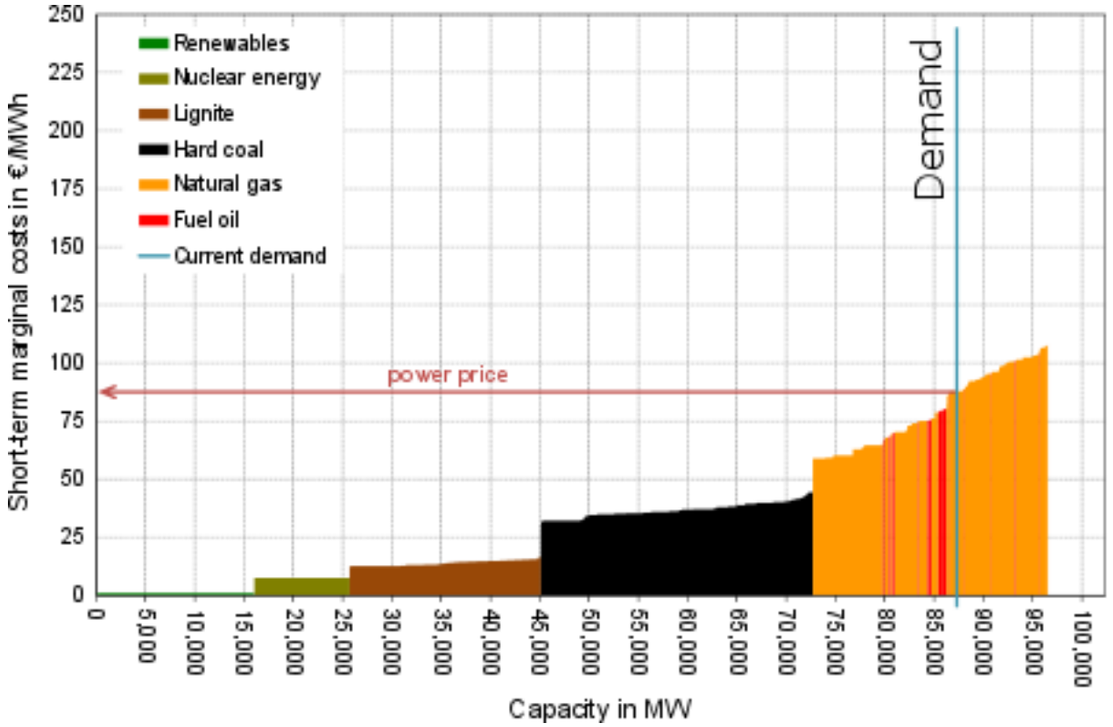
Shadow prices example

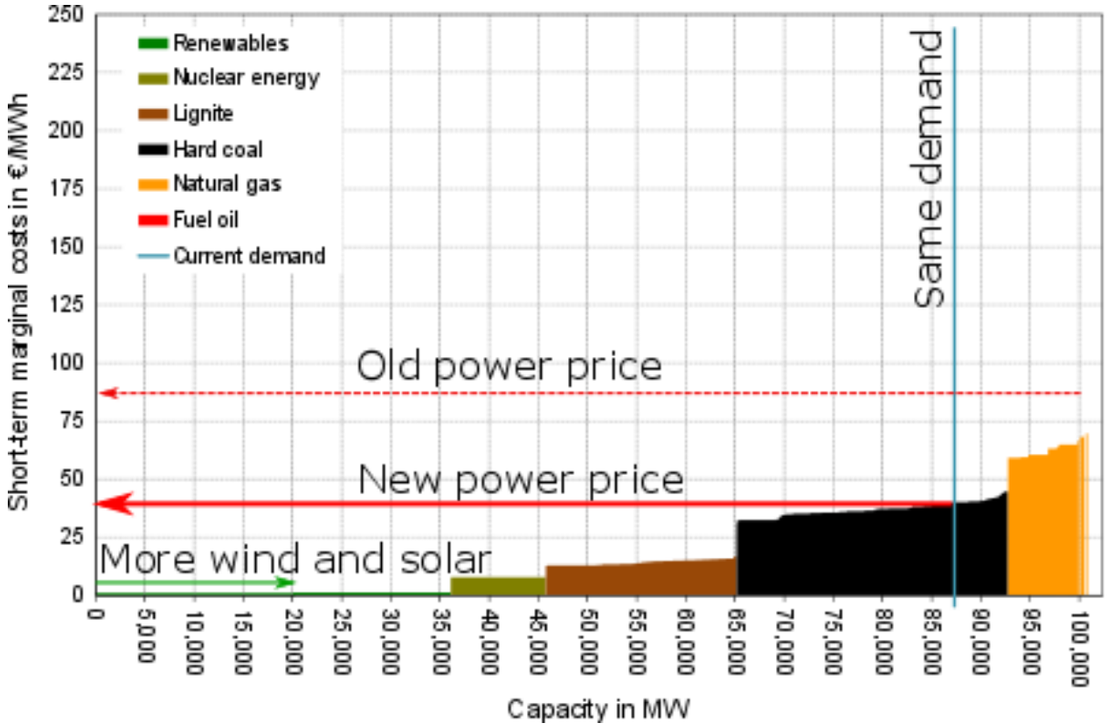
- 1 MW demand increase in summer leads to production increase from the cheapest available generator
- How to calculate the shadow price?
  - In this case, only option in short term is gas.
  - Therefore, the variable cost of gas sets overall electricity price.



Effect of low cost solar and wind

- More wind and solar power with zero-variable cost shift the supply curve to the right
- The electricity price decreases during the hours of wind and solar power production





- The revenue of wind and solar plants shrinks
- The marginal generator changes (natural gas & coal)

Revenue

With the power production p_t and the electricity prices (shadow prices) πel_t we can calculate the revenue of any plant in the system:

$$Revenue\ [CHF] = \sum_t (p_t[MW] * w[h] * \pi el_t[CHF/MWh])$$

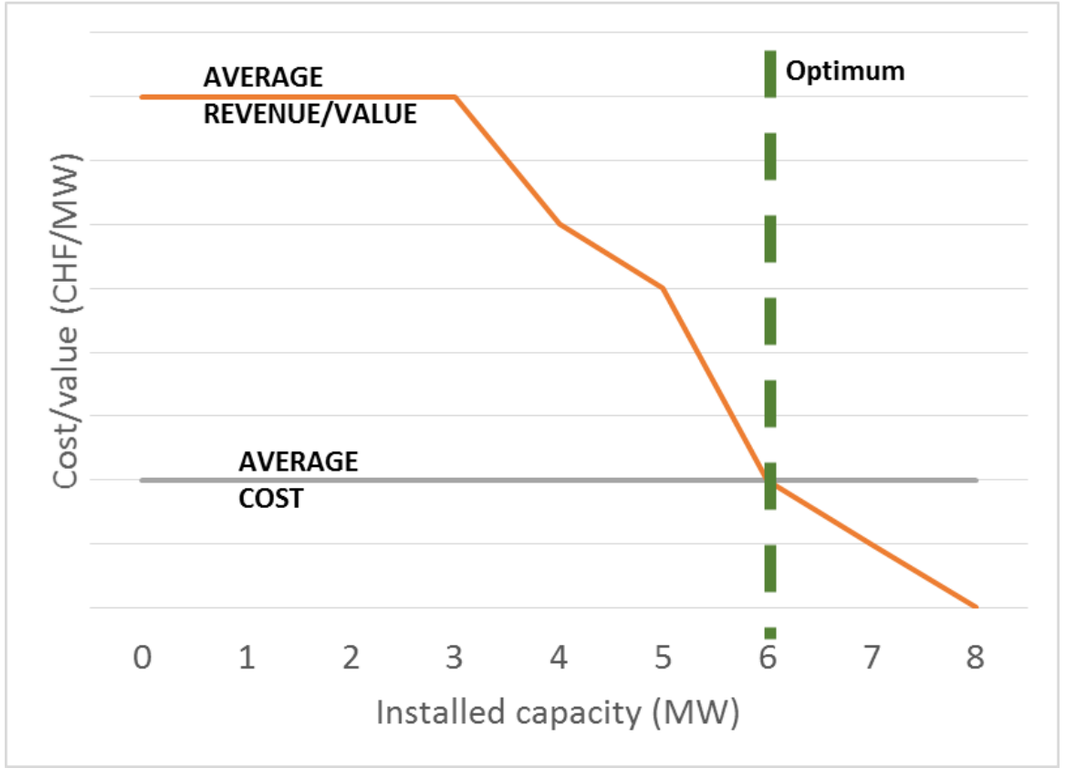
With w[h] the time slot weight (number of hours in time slot, e.g. 2190 hours in a season)

Cost is equal to revenue

- In linear optimization models, the total **revenue** of optimized capacity is equal to the **cost**, therefore:
  - Optimized components have net value equal zero
  - Optimized components don’t produce profits or losses

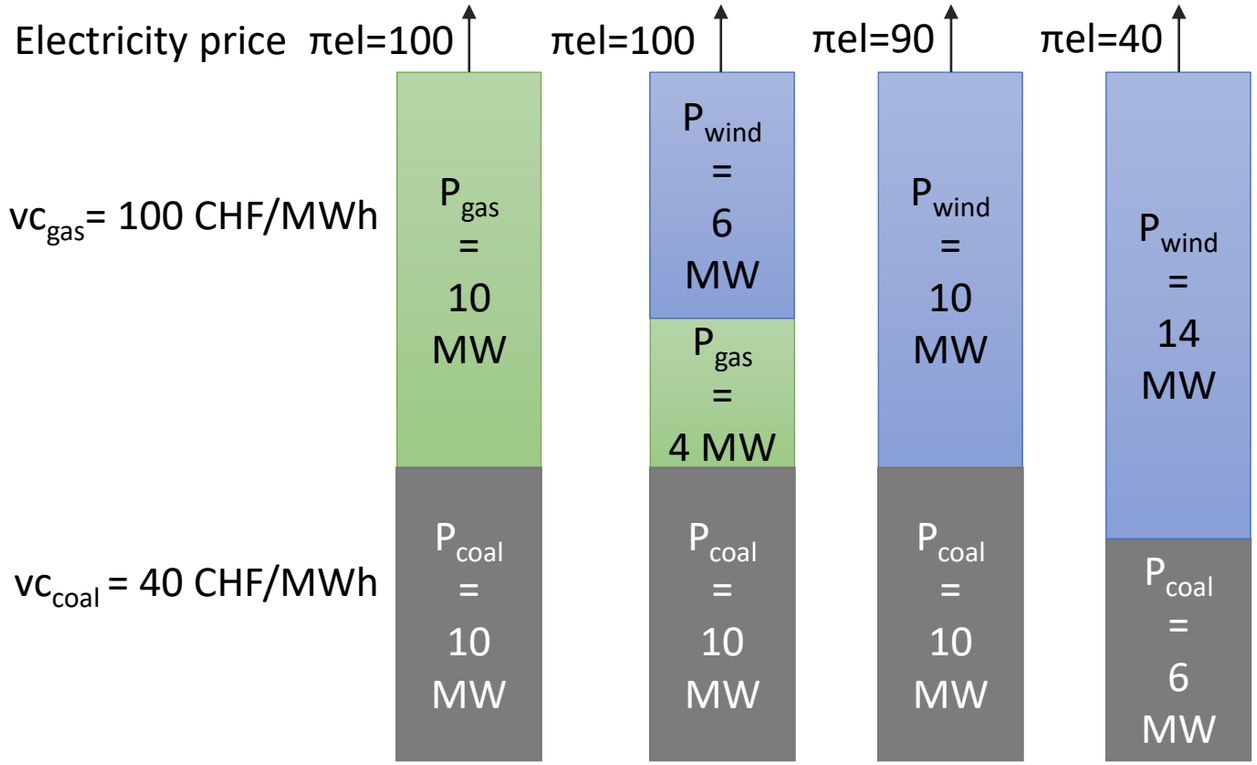
Example: Solar power

- Higher capacity of the same solar resource **reduces the electricity prices** at noon
- The **average revenue of solar power is reduced** if more solar power is added to the system
- When value does not equal cost:
  - Profit from PV means we can **lower system cost by adding more PV**
  - PV costs exceeding revenue means **we have installed too much PV**



Example: optimum capacities

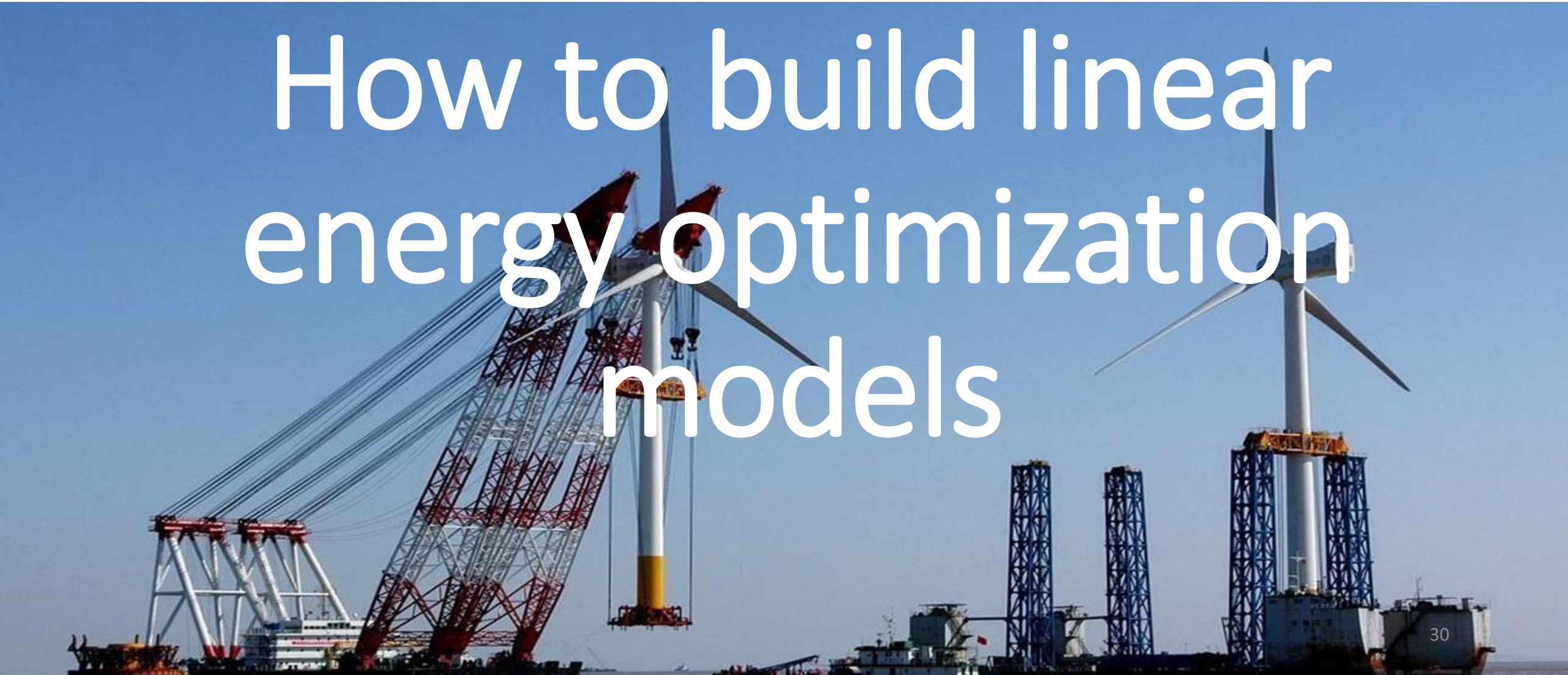
- Wind energy cost lower than gas $LCOE_{wind}$ = 90CHF/MWh
- Can replace gas power, but not cheaper coal power
- How much wind power would we install?
- What would the total cost be?
- The total CAPEX and revenue of wind power?
- What changes if the wind capacity is not optimal?



Total cost (CHF/h)	100*10 + 40*10 = 1400	1340	**1300** optimum!	1500
Wind revenue (CHF/h)	-	100*6	90*10	40*14
Wind cost (CHF/h)	-	90*6	90*10	90*14
Revenue - Value	**-**	**60**	**0**	**-700**

Part III:

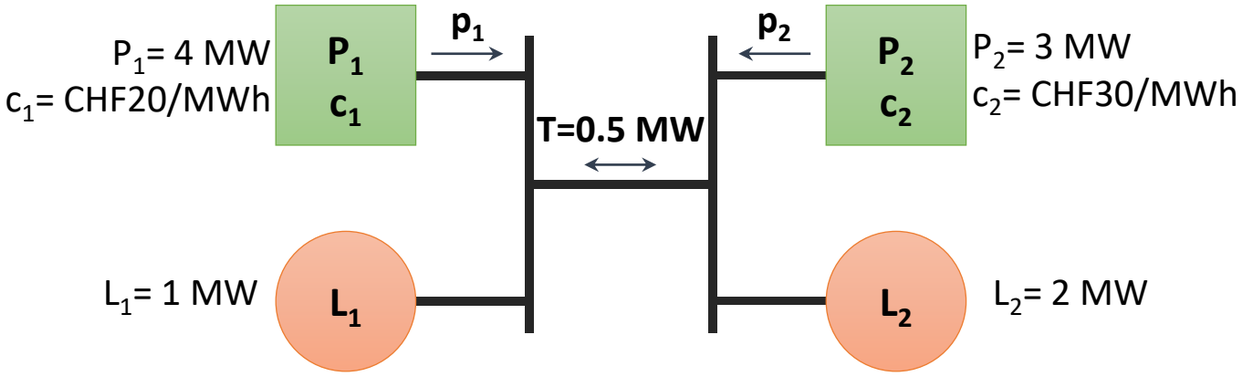
How to build linear energy optimization models



Linear optimization models

- Linear optimization models consist of
  - Sets
  - Parameters
  - Variables
  - Constraints
  - The objective function
- Mathematically, linear optimization models can be expressed as a single matrix

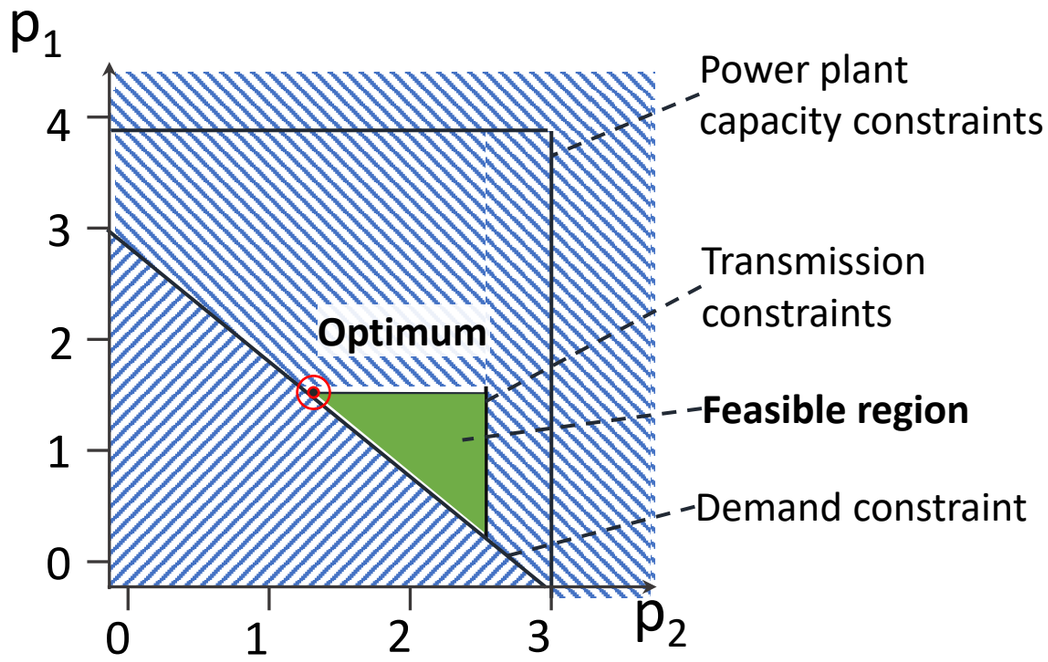
Example: Graphical optimization of a power system



Aim: find values of $\mathbf{p_1}$ and $\mathbf{p_2}$ such that the total cost is minimized

Demand constraint $p_1 + p_2 \geq L_1 + L_2 = 3$ MW

Objective function $20p_1 + 30p_2$



Power plant capacity constraints
$p_1 \leq P_1 = 4$ MW
$p_2 \leq P_2 = 3$ MW

Transmission constraints
$p_1 \leq L_1 + \mathrm{T} = 1.5$ MW
$p_2 \leq L_2 + \mathrm{T} = 2.5$ MW

General linear optimization model

Power plant capacity constraints
$p_1 \leq P_1 = 4\ \text{MW}$
$p_2 \leq P_2 = 3\ \text{MW}$

Transmission constraints
$p_1 \leq L_1\text{+T} = 1.5\ \text{MW}$
$p_2 \leq L_2\text{+T} = 2.5\ \text{MW}$

Demand constraint
$p_1 + p_2 \geq L_1\text{+}L_2 = 3\ \text{MW}$

Objective function
$20p_1 + 30p_2$

Canonical matrix form (linear program):

$$\text{minimize} \left[\begin{pmatrix} c_1 & c_2 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \right]$$

$$\text{with} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ -1 & -1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \leq \begin{pmatrix} 4 \\ 3 \\ 1.5 \\ 2.5 \\ -3 \end{pmatrix}$$

$$\text{minimize} \quad \boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$$

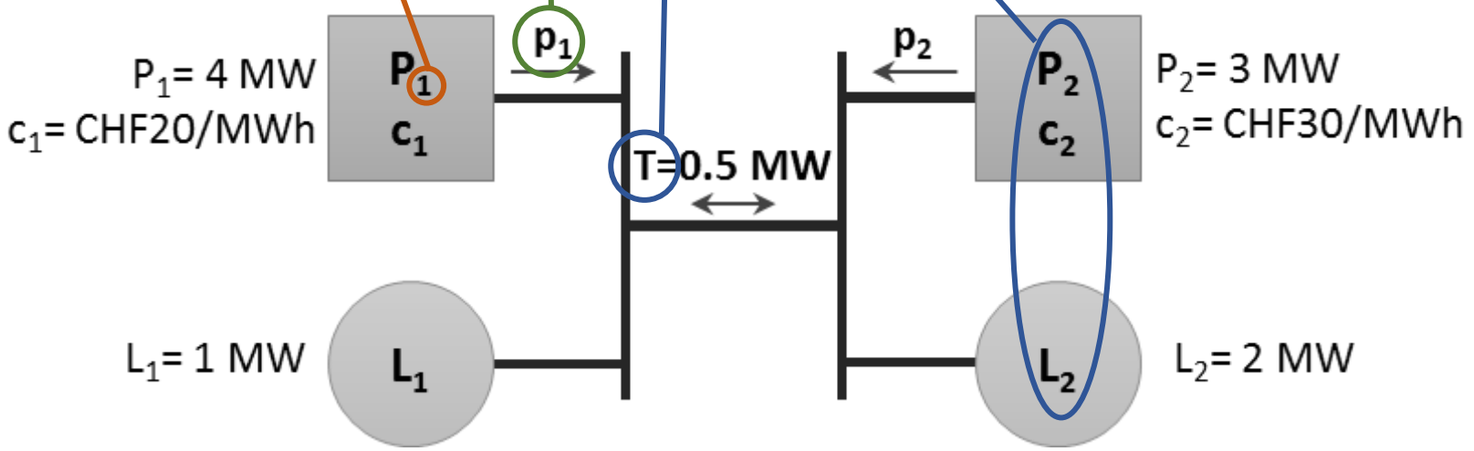
$$\text{subject to} \quad A\boldsymbol{x} \leq \boldsymbol{b}$$

Structure of an optimization model

Variables (what's being optimized)
In this case: power production p_i

Parameters (input data); in this case:
- Capacity (P_i)
- Variable cost (c_i)
- Transmission capacity (T): no index i
- Load L_i

Sets (indices)
In this case: Regions i {1,2}



Objective function (what's being minimized/maximized):
In this case: minimize total cost

Constraints: Equations/inequalities limiting the variable choice

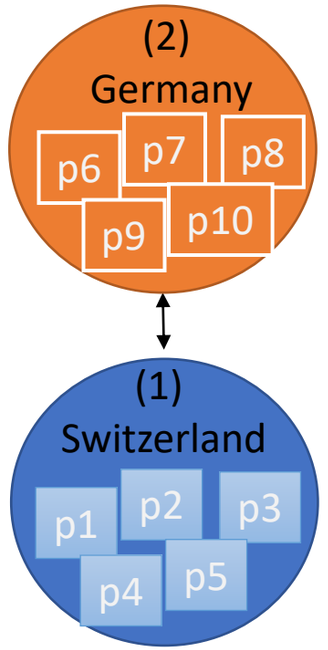
In the exercise: Pyomo

- Pyomo is a python package for building optimisation models
- Can be used in many different contexts
- We write a special Python description of the problem
- Uses a *solver* internally the performs the optimisation (e.g. CPLEX)

- http://www.pyomo.org/

Structure of an optimization model: *Sets*

- What do the parameter/variables/constraints depend on?
  - **Nodes (countries, regions, substations, buildings…)**
  - **Power plants** (e.g. installed capacity)
  - **Hour/time slot** (e.g. demand *parameter* defined for each hour/time slot, power plant production *variable* defined for each hour/time slot)
  - **Fuels** (e.g. fuel price defined for each fuel, emission intensity defined for each fuel)
  - **Etc..**



Example of node Sets and power plant sets

Structure of an optimization model: *Sets*

- Power plants and 4 seasons
- Sets are attributes (properties) of the model object `m`
- Sets are “lists of names”

```
m = po.ConcreteModel()
m.seasons = po.Set(initialize=['0_spring', '1_summer’,…])
m.power_plants = po.Set(initialize=['solar','gas‘, ‘wind’,…])
```

- Multiplying sets gives “all combinations” (all power plants during all seasons) :

```
m.seasons * m.power_plants
```

- subset, e.g. wind and solar:

```
m.wind_solar= po.Set(within=m.power_plants,
                     initialize=['solar',‘wind'])
```

Structure of an optimization model: *Parameters*

- Parameters are input data
  - Costs
  - Capacities of power plants
  - Capacity factors
  - Transmission capacities
  - Demand
  - Efficiency
  - Time step duration

Structure of an optimization model: *Parameters*

- Just like sets, parameters are attributes of the model object `m`
  - e.g. capacity factors m.cf defined for all VRE plants and all seasons (m.power_plants * m.seasons)

```
m.cf = po.Param(m.power_plants * m.seasons,
                initialize={('wind', '0_spring'): 0.175,
                            ('wind', '1_summer'): 0.131, … etc.})
                            ('solar', '1_spring'): 0.09, … etc.})
```

- e.g. power plant capacity defined for all power plants

```
m.capacity_old = po.Param(m.power_plants,
                          initialize={'wind': 20000, 'solar': 15000})
```

- e.g. time slot weight/duration defined for all time slots (seasons)

```
m.season_weight = po.Param(m.seasons, initialize={'0_spring': 2190})
```

Structure of an optimization model: *Variables*

- E.g. produced power, installed capacity, stored energy, …
- Aim: find variable values to minimize the total cost/minimize emissions/maximize profit, etc.
- Variables have
  - Bounds (-∞, ∞), (0,+∞), etc.
  - Domains (real numbers, binaries, integer values)

Structure of an optimization model: *Variables*

In the exercise

- Power production each season + new capacity for some plants
- All variables are positive (0,+∞) and real numbers (default)

```
m.pwr = po.Var(m.power_plants * m.seasons,
               bounds=(0, None))
```

Structure of an optimization model: *Constraints*

- Need to set constraints on the possible solutions
- Defined as:
  - Equality constraints (e.g. energy balance)
  - Inequality constraints (e.g. power generated less than maximum power capacity)

Structure of an optimization model: *Constraints*

- Energy balance constraints: for each time step, the supply is greater or equal the demand for each time step t (e.g. season)

$p_{nuclear,t} + p_{gas,t} + p_{hydro,t} + p_{coal,t} + p_{solar,t} >= D_t$

- Function `demand_constraint_equation` is called for each season and returns an inequality “supply >= demand”

```
    def demand_constraint_equation(m, season):
        return (sum(m.pwr[plant, season] for plant in m.power_plants) >=
m.demand[season])
```

- Constraint “for each season”

```
    m.demand_constraint = po.Constraint(m.seasons,
                                        rule=demand_constraint_equation)
```

Constraints

Pass the ‘model’ to the function so we can access the variables

Sum of power production of all power plants during a given season.
note: sum of all power plants is always `sum(something[plant] for plant in m.power_plants)`

Constraints constructed for each single season (4 seasons in total, therefore 4 single constraints in total)

```
def demand_constraint_equation(m, season):
    return (sum(m.pwr[plant, season] for plant in m.power_plants) >=
            m.demand[season])


m.demand_constraint = po.Constraint(m.seasons,
                                    rule=demand_constraint_equation)
```

“for each season” expressed through the model sets previously defined

Constraints: Variable new capacity

What if we allow for new capacity investments for a set of power plants `m.new_power_plants`?

⇒Variable `m.cap_new` for new plants

```
def capacity_constraint_equation(m, plant, season):
    if plant in m.new_power_plants:
        return m.pwr[plant, season] <= m.cap[plant] + m.cap_new[plant]
    else:
        return m.pwr[plant, season] <= m.cap[plant]


m.capacity_constraint = po.Constraint(m.power_plants * m.seasons,
                                      rule=capacity_constraint_equation)
```

Parameter exists for all plants

Variable only exists for new power plants

Constraints: Profile constraint

- For each time slot and each wind or solar plant, the produced power must be equal the installed capacity times the capacity factor $\mathrm{cf}_{\mathrm{plant,t}}$:

$$p_{\mathrm{plant},t} = \mathrm{cf}_{\mathrm{plant},t} \cdot P_{\mathrm{plant}} \text{ for each variable renewable energy plant}$$

- The capacity factor follows from the resource availability
  - e.g. zero at night for solar, wind produces more power in winter as compared to summer

Structure of an optimization model: *Objective function*

- Must produce a single value
  - Examples: Total cost (minimize), emissions (minimize), revenue/profit (maximize)
  - Total cost:
    - Fuel cost of all power plants
    - Fixed and variable OPEX, CAPEX
    - Others costs like ramping costs, start-up costs

Structure of an optimization model: *Objective function*

Example: Yearly cost from Variable Cost (VC) and Fixed Costs (FC)

$$\mathrm{VC}_{\mathrm{fuel,plant},t} = \frac{\left(\mathrm{vc}_{\mathrm{fuel(plant)}} + i_{\mathrm{CO_2,fuel(plant)}} \cdot \pi_{\mathrm{CO_2}}\right)}{\eta_{\mathrm{fuel}}} \cdot p_{\mathrm{plant},t} \cdot w_t$$

$$\mathrm{FC}_{\mathrm{plant}} = \left(\mathrm{fc}_{\mathrm{OPEX}} + \alpha \cdot \mathrm{fc}_{\mathrm{CAPEX}}\right) \cdot P_{\mathrm{i}}$$

$$\text{Total Cost} = \sum_{\mathrm{plant}} \left\{ \sum_{t} \left[\mathrm{VC}_{\mathrm{fuel,plant},t}\right] \right\} + \sum_{\mathrm{plant}} \mathrm{FC}_{\mathrm{plant}}$$

$vc \left[\frac{EUR}{MWh_{fuel}}\right]$

$i_{CO2} \left[\frac{t_{eqCO2}}{MWh_{fuel}}\right]$

$\pi_{CO2} \left[\frac{EUR}{t_{eqCO2}}\right]$

$\eta_{\mathrm{fuel}}[\frac{MWh_{el}}{MWh_{fuel}}]$

$p[MW_{el}]$

$w \left[\frac{h}{year}\right]$

Objective function: Formulation in Pyomo

Note: double sum (all seasons, all power plants)

```
TOT_VAR = sum(m.season_weight[season] * m.pwr[plant, season]
              * (m.fuel_cost[plant] + m.price_co2 * m.co2_intensity[plant])
              / m.eff[plant]
              for plant in m.power_plants for season in m.seasons)

TOT_FIX = sum(m.cap_new[plant] * m.fixed_cost[plant]
              for plant in m.new_power_plants)

m.obj = po.Objective(expr=TOT_VAR + TOT_FIX, sense=po.minimize)
```

Total cost based on components defined above

Define whether we minimize or maximize

The full example in Pyomo

Sets

```
m.regions = po.Set(initialize=['one', 'two'])
```

Variables

```
m.power = po.Var(m.regions, bounds=(0,None))
```

Parameters

```
m.capacity = po.Param(m.regions, initialize={'one': 4, 'two': 3})
m.demand = po.Param(m.regions, initialize={'one': 1, 'two': 2})
m.cost = po.Param(m.regions, initialize={'one': 20, 'two': 30})
m.transmission_cap = po.Param(initialize=0.5)
```

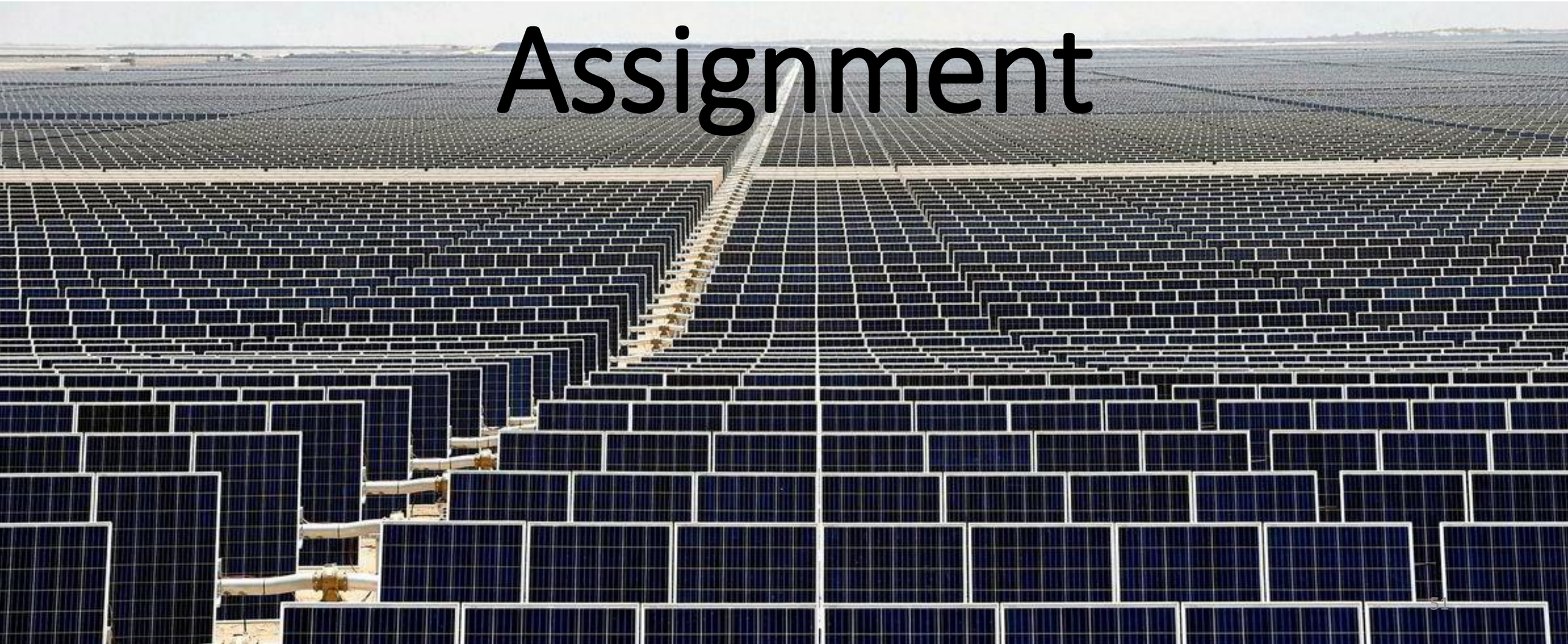
Constraint

```
def transmission_constraints_rule(m, region):
    return m.power[region] <= m.demand[region] + m.transmission_cap
m.transmission_constraints = po.Constraint(m.regions, rule=transmission_constraints_rule)

def demand_constraint_rule(m):
    return sum(m.power[region] for region in m.regions)
           >= sum(m.demand[region] for region in m.regions)
m.demand_constraint = po.Constraint(rule=demand_constraint_rule)

def capacity_constraint_rule(m, region):
    return m.power[region] <= m.capacity[region]
m.capacity_constraint = po.Constraint(m.regions, rule=capacity_constraint_rule)
```

Objective

```
def objective_rule(m):
    return sum(m.power[region] * m.cost[region] for region in m.regions)
m.objective = po.Objective(rule=objective_rule, sense=po.minimize)
solver.solve(m, tee=True)
```

Part IV: Assignment



Purpose

- Learn how to build a simple (but scalable) energy optimization model to represent the national electricity of a region
- How does the CO2 price affect the optimal system composition?
- What happens to the wind revenue if we install more and more wind capacity?
- Analyze the effect on shadow prices/electricity prices

Purpose

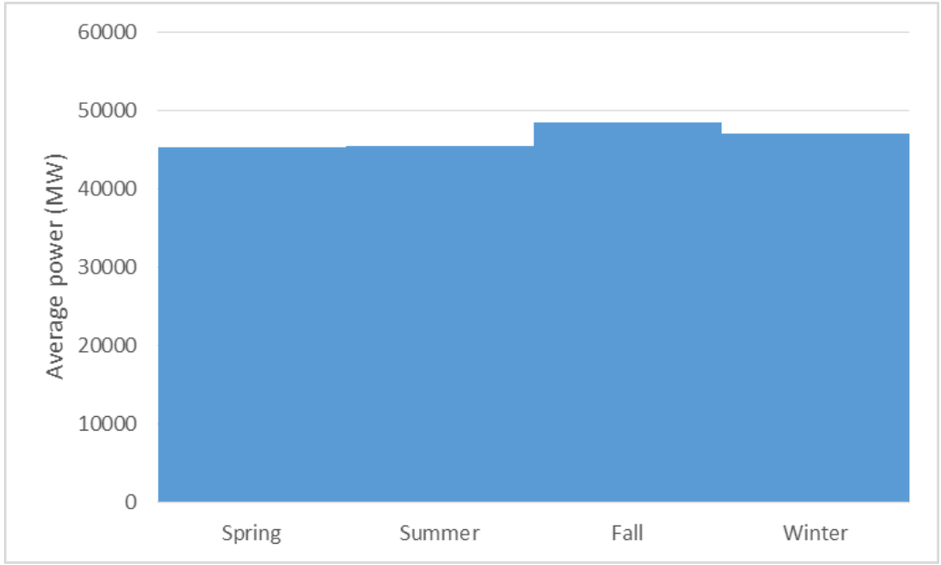
Optimize operation

- Nuclear
- Coal
- Gas
- Wind
- Solar PV

Optimize operation **and** capacity

- **New** solar PV
- **New** wind
- **New** gas power plants

Cover demand at lowest cost



Classification of the model

- Optimization of operation and capacity expansion
- Fully deterministic (perfect foresight all seasons)
- Dispatch model with aggregated power plants (e.g. all nuclear power plants optimized as one)
- Suited to analyze the impact of policy measures on drivers/optimal system composition

System:

- Single year (annualized CAPEX), four time slots (average seasons: spring, summer, fall, winter)
- Cost minimize the operation of 5 plants and invest in 3 new plant types
- Based on the German power system

Method I

- The exercise model is formulated in *Pyomo* (Python optimization module)
- We use Jupyter Notebooks (similar to *Monte Carlo Simulation and Techno-economic analysis*)
- The exercise is not about programming Python but formulating optimization models
    - We use Python/Pyomo as a modelling environment; therefore only few commands and concepts are necessary, the rest can be ignored/will be provided to you

Method II

- The analysis will be performed in Excel
  - The most advanced Excel function you need to perform the analysis and to reshape the data is “SUMIFS” (“SOMME.SI.ENS in French); please get yourself familiar with it (see the SUMIFS example sheet in the Excel report template file)
  - Of course you can use any other Excel approach for the analysis (pivot tables, INDEX(…, MATCH(…)), array formulas, etc.)
- Use Word template for the report
- **Detailed instructions and notes are included with the Jupyter Notebook. Please make sure to read them.**

Questions?