Week 9

Energy System Optimization

Linear Optimization for Energy System Planning and Analysis

Martin C. Soini, Arthur Rinaldi

Course "**Methods for technical and economic energy analysis**"

07.05.2020

UNIVERSITÉ DE GENÈVE

Learning goals

UNIVERSITÉ DE GENÈVE

- 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 CO_2 emissions has the potential to impact the cost-effective energy transition

Agenda

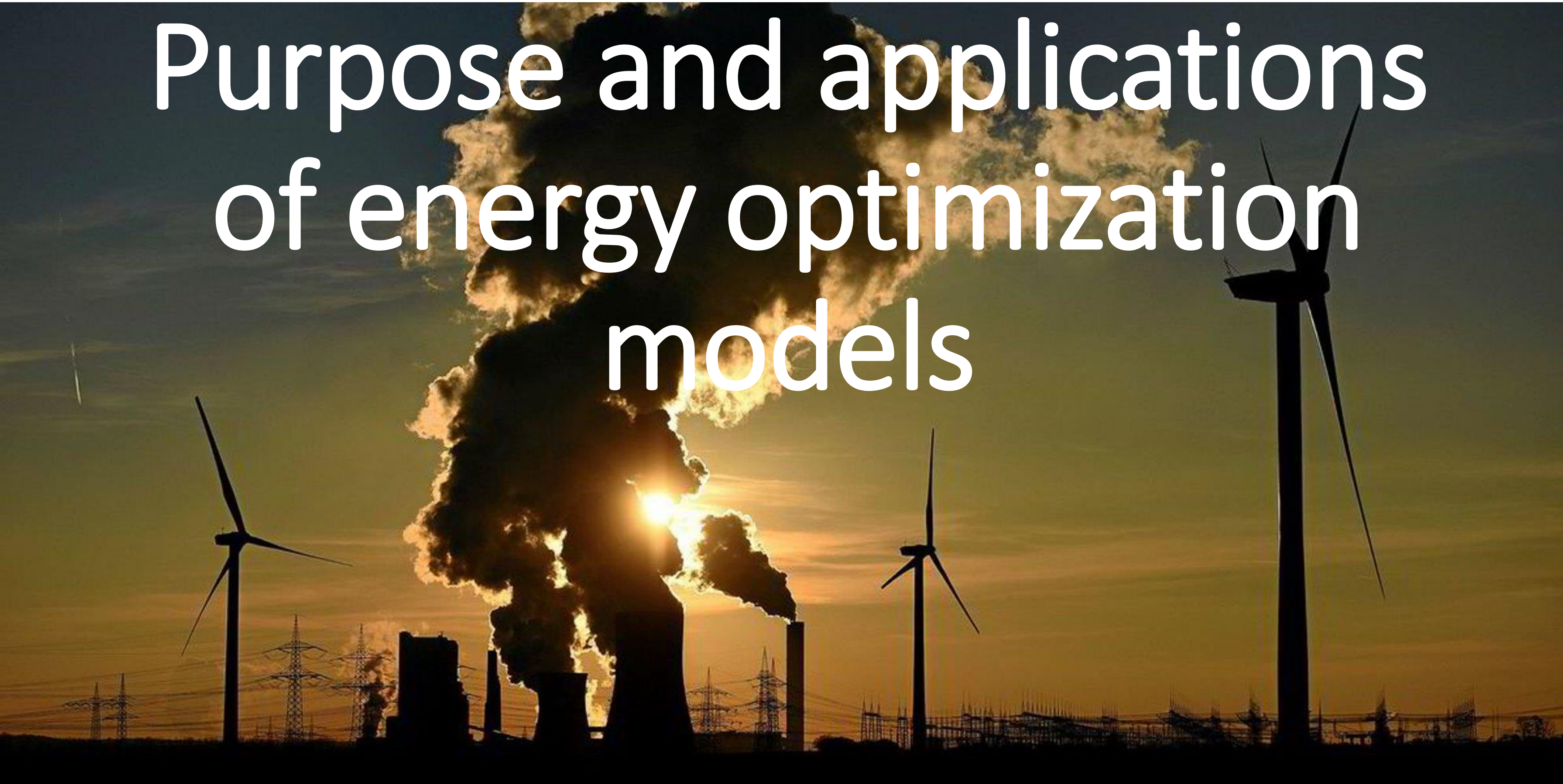
UNIVERSITÉ DE GENÈVE

- **Part 1: Purpose and applications of energy optimization models**
  - Examples from research
  - Types of models
- **Part 2: How to build energy optimization models**
  - Introductory example
  - Structure of an optimization model
  - The introductory example formulated in Pyomo/Python
- **Part 3: Various important concepts**
  - Capacity factor (reminder)
  - Annualized cost and levelized cost (reminder)
  - Shadow prices, electricity prices and the marginal generator
  - Revenue
  - The merit order effect
  - Revenue=cost in linear systems
- **Part 4: Assignment notes**

UNIVERSITÉ DE GENÈVE

Part I:

Purpose and applications of energy optimization models



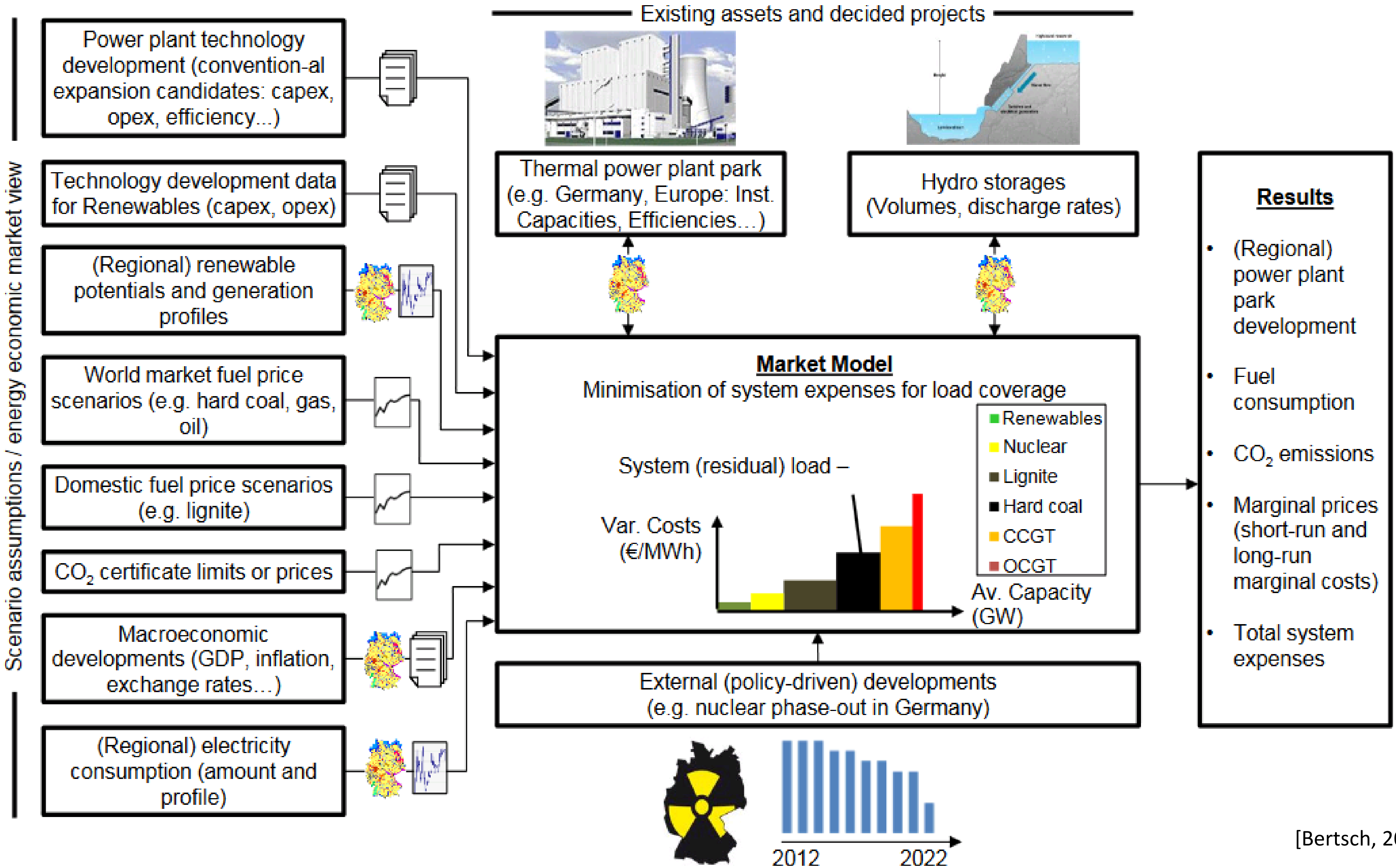
Optimization

- 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)
- ….and which are subject to certain system **constraints** (supply=demand, cannot produce 2GW with a 1GW power plant, etc)

Integrated resources planning

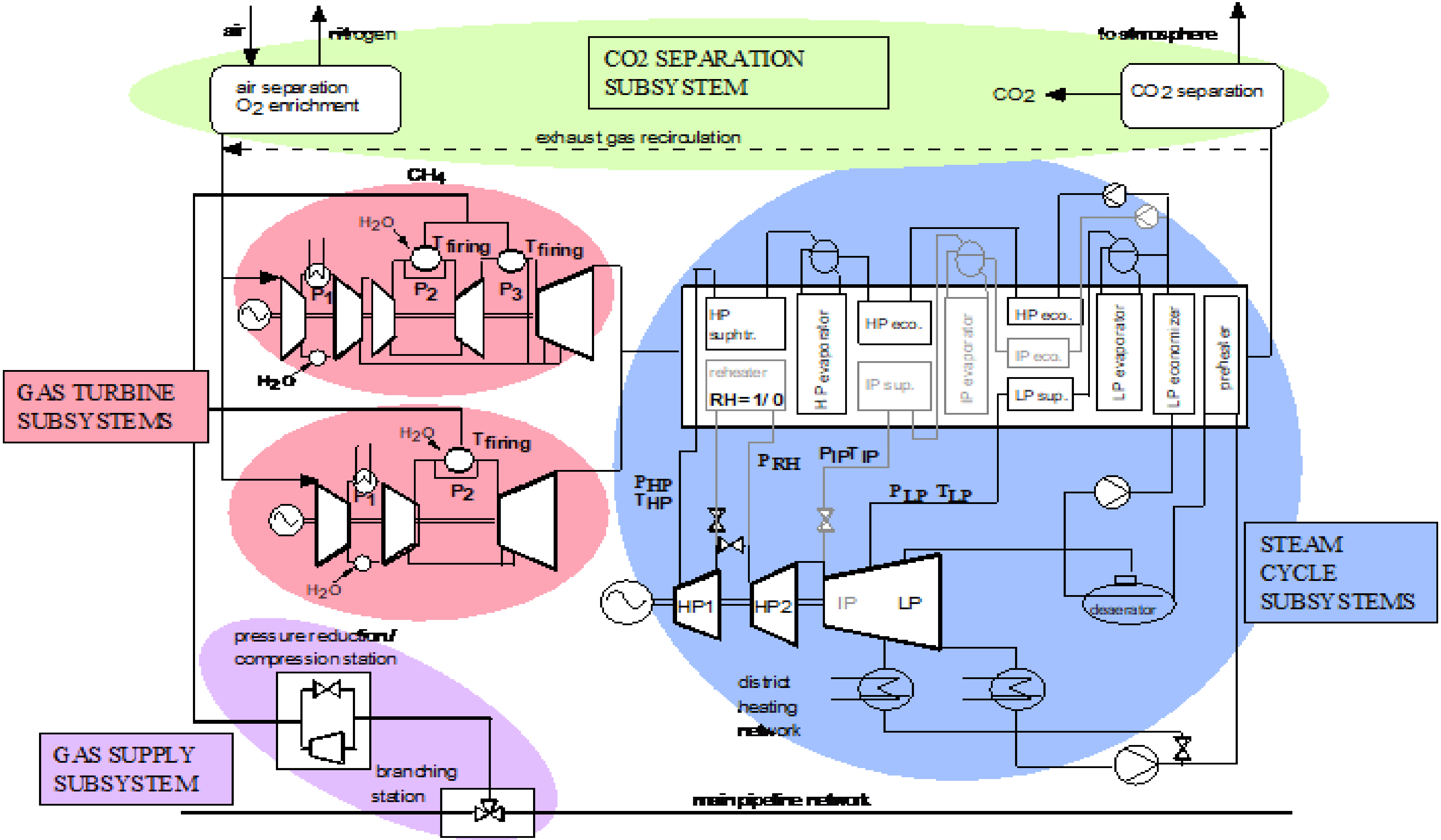
UNIVERSITÉ DE GENÈVE

Input and Output of an Energy Systems Model



[Bertsch, 2015]

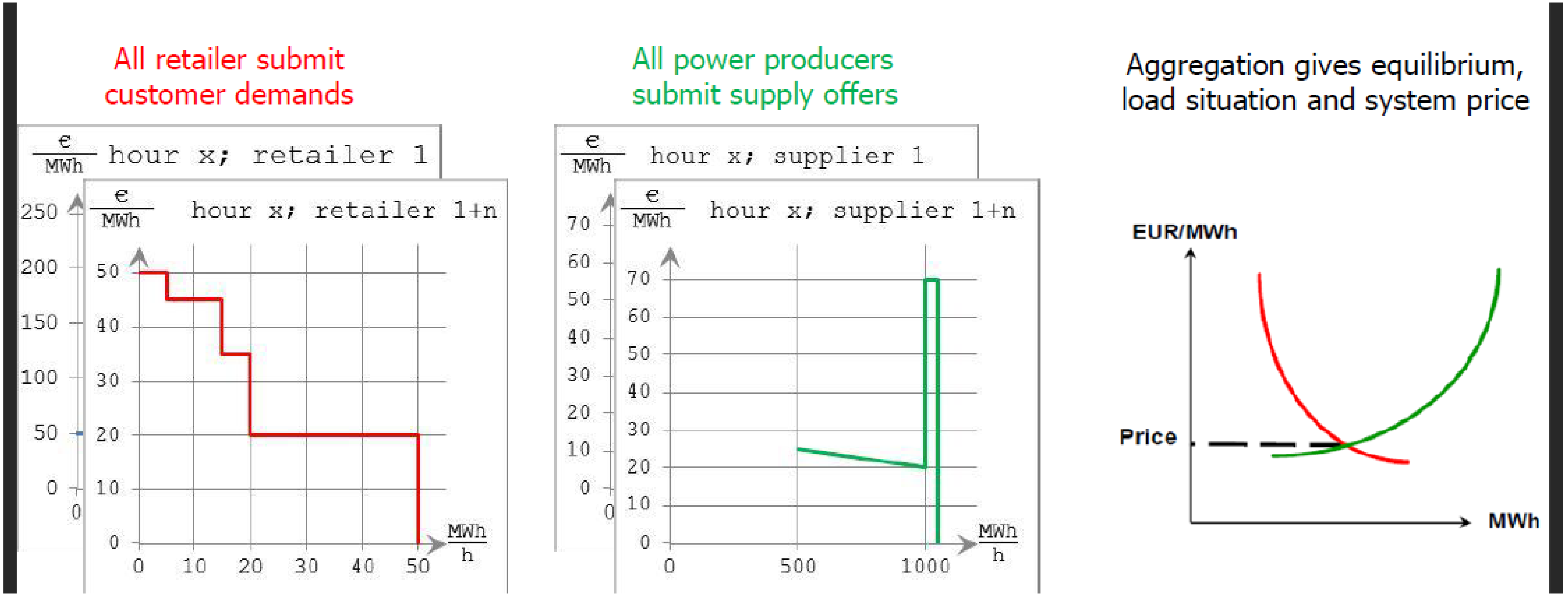
Integrated energy systems



[Pelster et al., 2003]

Markets

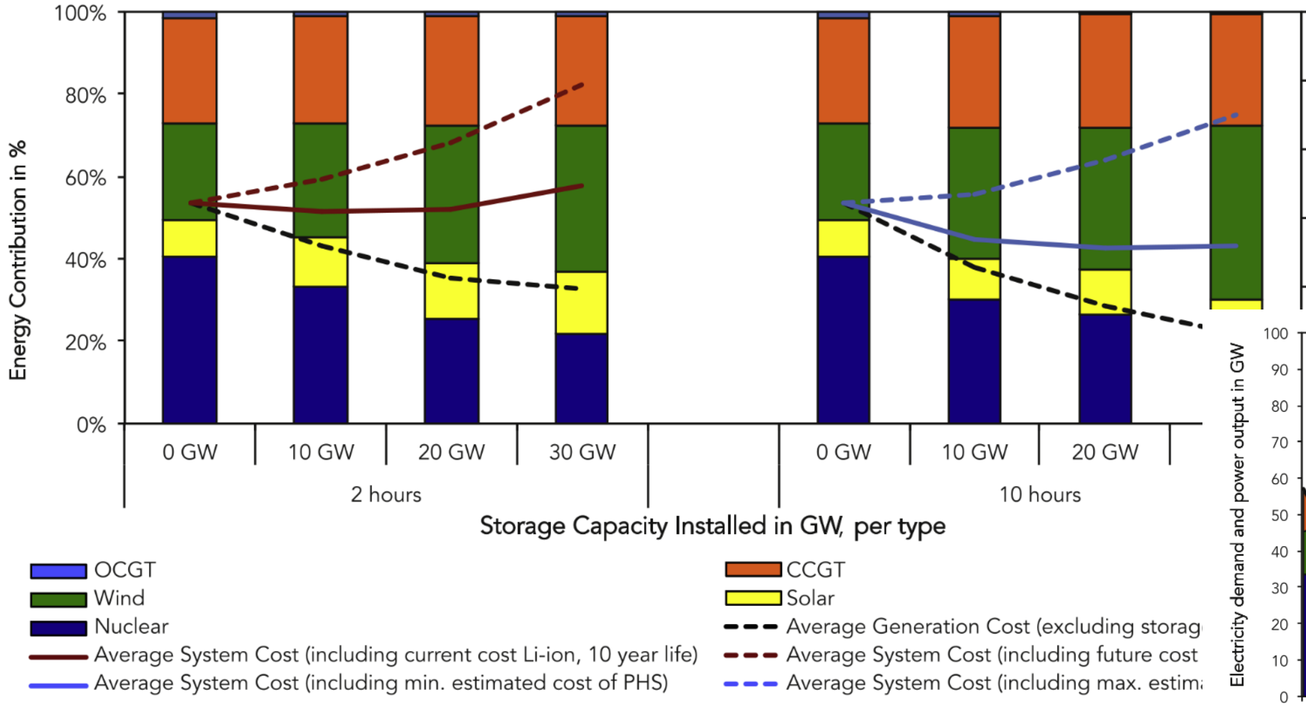
- Markets minimize costs to cover demand
- Optimization can be used to approximate market behavior
- Answer questions like:
  - How does the market react if we change certain parameters?
  - Estimate future electricity prices
  - Analyze drivers
  - Controlled parameter changes to understand market behavior

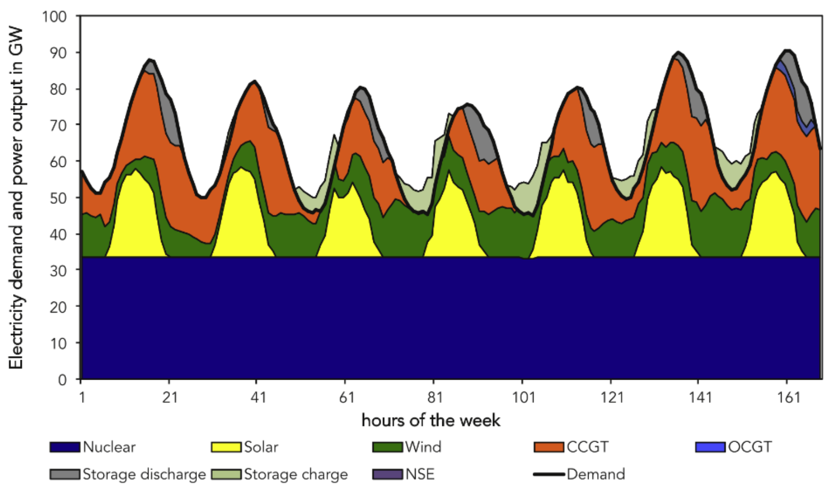


Examples from research 1: Optimal system configuration

UNIVERSITÉ DE GENÈVE

- *The value of energy storage in decarbonizing the electricity sector*
- http://dx.doi.org/10.1016/j.apenergy.2016.05.014
- Approach: Unit commitment model with investments, approximating the year through 4 representative weeks; assumes no pre-existing capacity; Texas system
- How does the optimal system composition/costs change if storage is added?

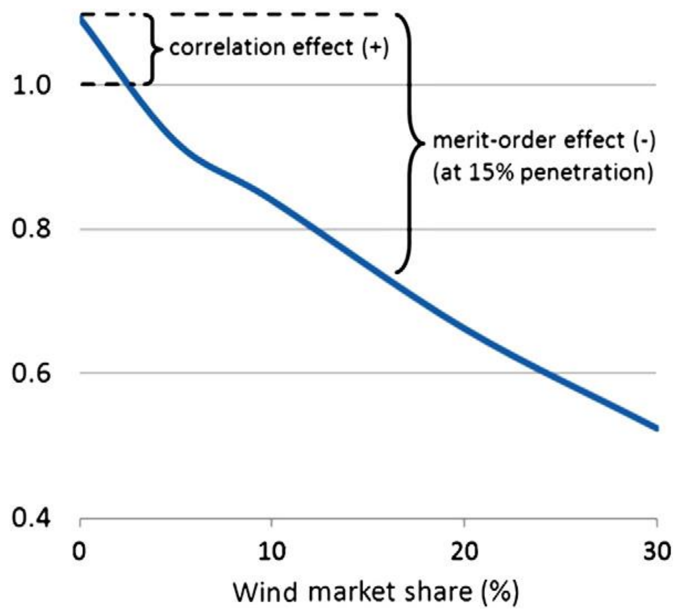


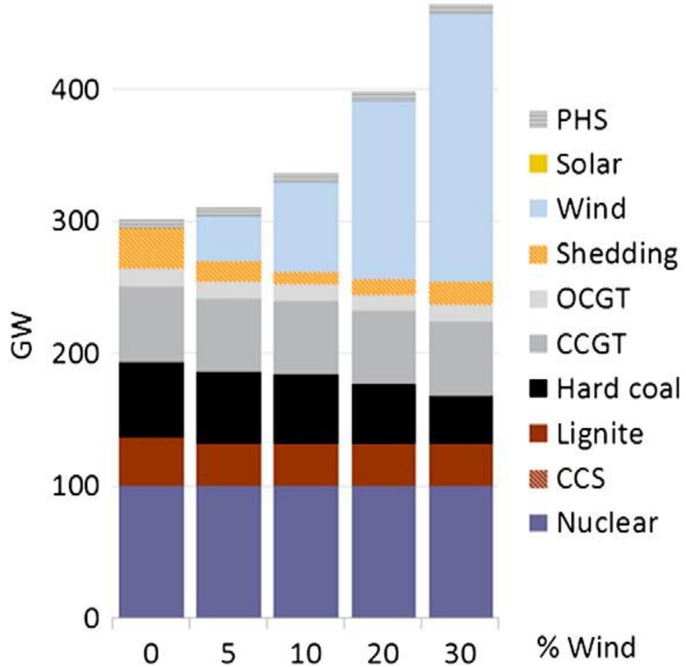


Examples from research 2: Value erosion of wind and PV

UNIVERSITÉ DE GENÈVE

- *The market value of variable renewables, The effect of solar wind power variability on their relative price*
- http://dx.doi.org/10.1016/j.eneco.2013.02.004
- Approach: 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 only half of that from a constant source of electricity."



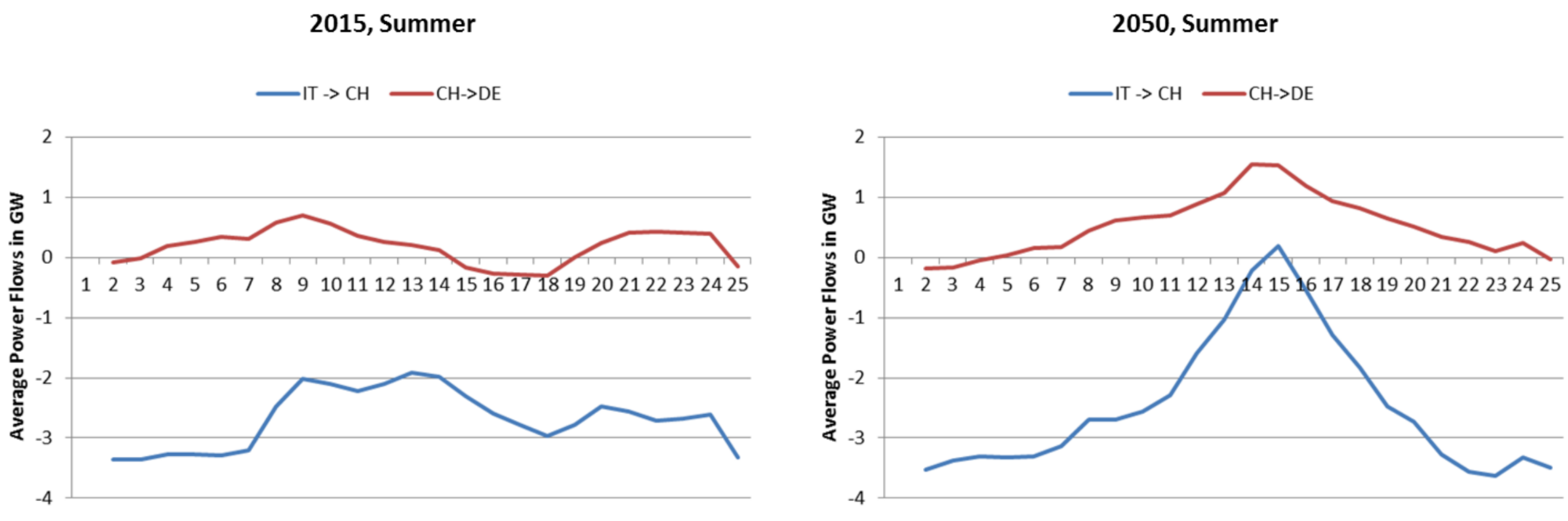


Examples from research 3: Future role of Switzerland

UNIVERSITÉ DE GENÈVE

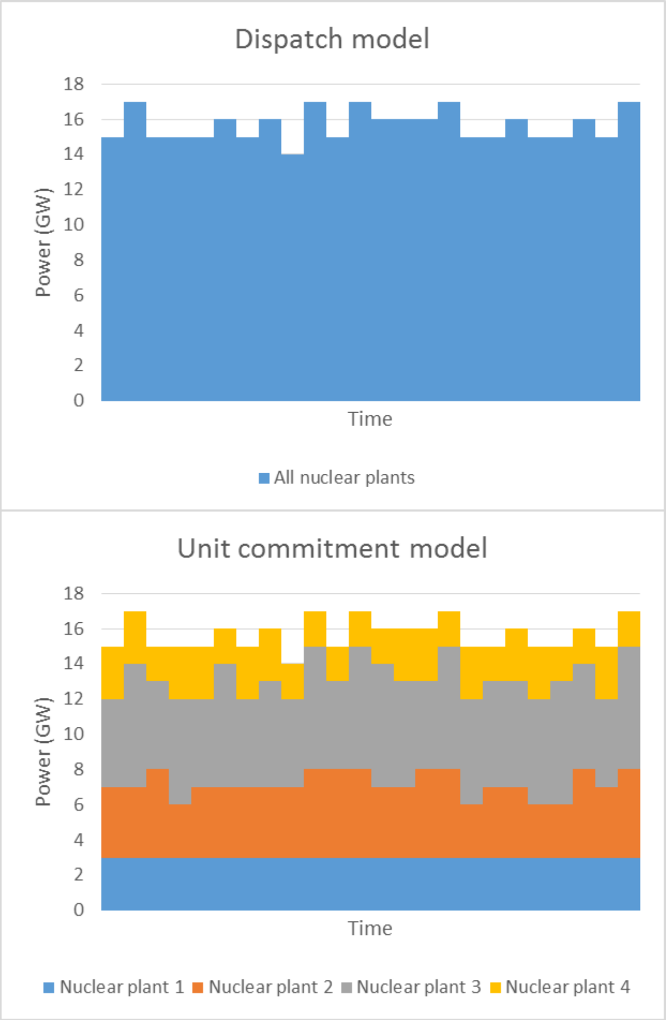
- *Linking Europe - The Role of the Swiss Electricity Transmission Grid until 2050*
- https://www.sccer-crest.ch/fileadmin/FILES/Publications/FoNEW_2014_03.pdf
- Approach: Detailed Swiss market model, all hydro power plants

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



Types of models (not comprehensive)

- Dispatch vs. unit commitment
  - *Assignment: typical simple dispatch model*
- Temporal representation: 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*
- Grid vs. “copperplate”
  - Assignment: copperplate



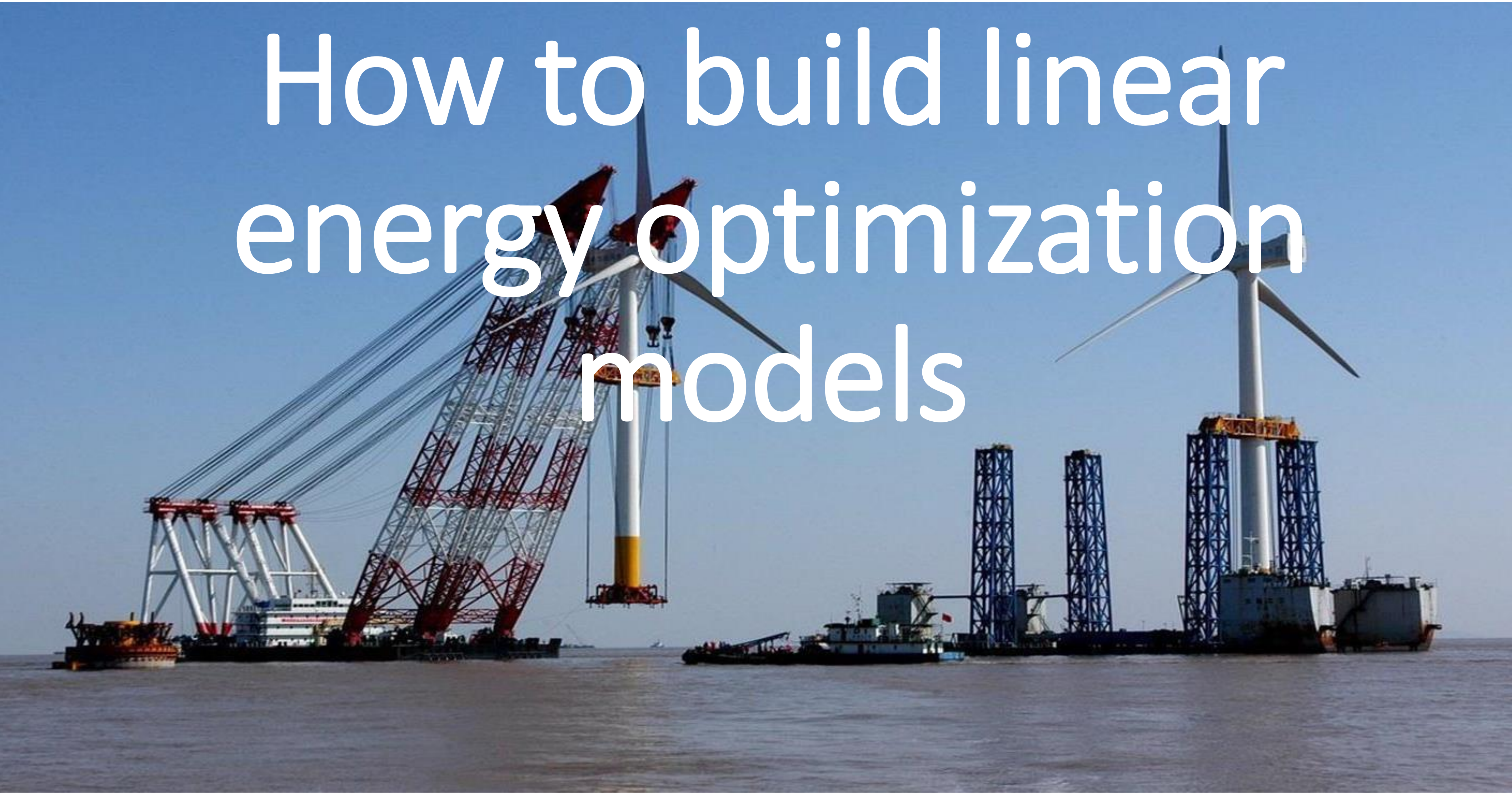
Part I: Purpose and applications

Summary

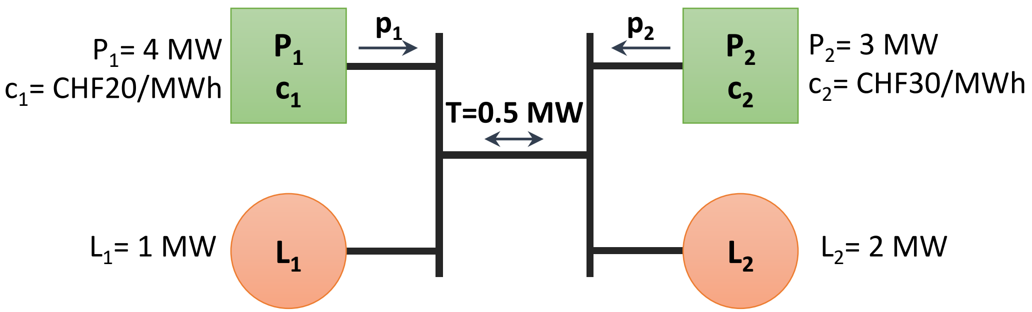
- Linear optimization models are useful to model a broad variety of energy systems
- The optimization allows to approximate perfect (electricity) markets
- The design of the model is determined by both the system we want to represent and the questions we want to answer

Part II:

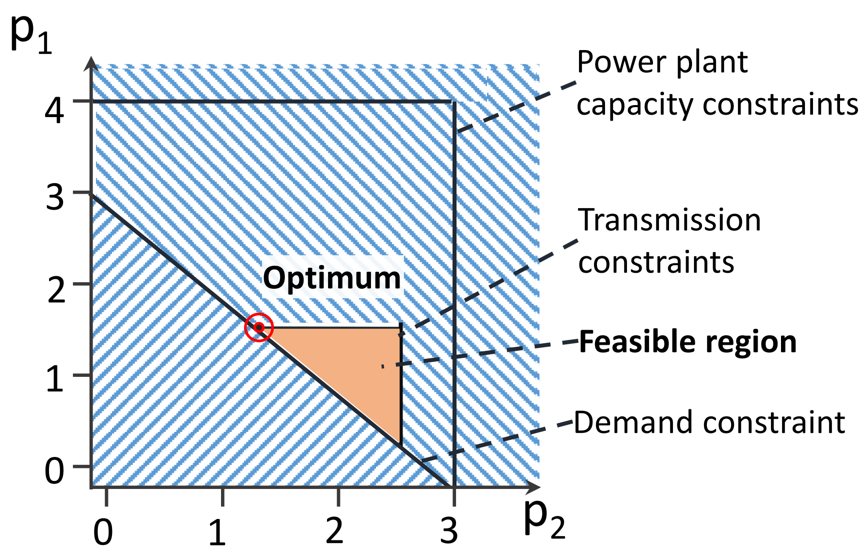
How to build linear energy optimization models



Example: Graphical optimization of a power system



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



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

Objective function
$$20p_1 + 30p_2$$

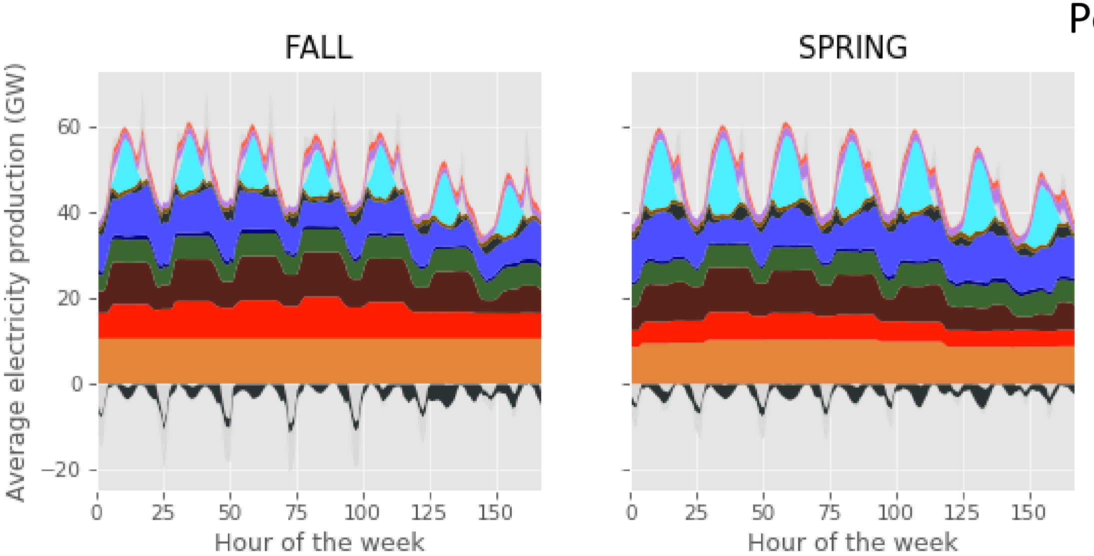
General linear optimization model

Canonical matrix form (all linear programs):

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

$$\text{with} \begin{pmatrix} -1 & -1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \end{pmatrix} \leq \begin{pmatrix} -3 \\ 1.5 \\ 2.5 \\ 4 \\ 2 \end{pmatrix}$$

minimize	$\boldsymbol{c}^{\mathrm{T}}\boldsymbol{x}$
subject to	$A\boldsymbol{x} \leq \boldsymbol{b}$



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

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

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

Objective function $20p_1 + 30p_2$

Structure of an optimization model

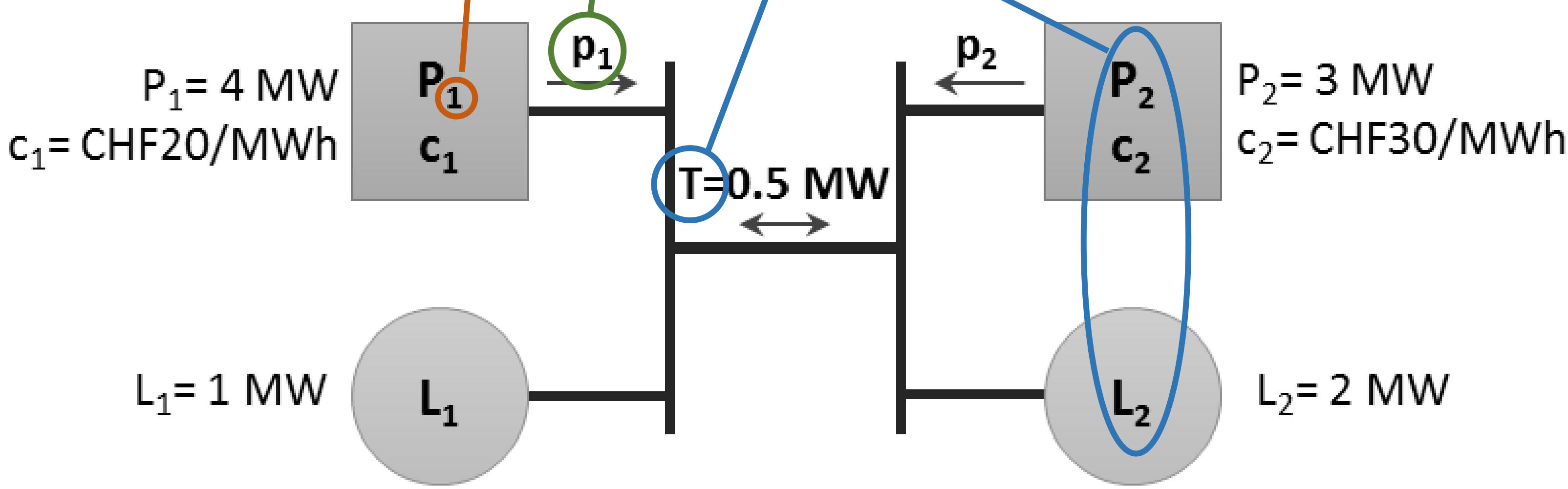
UNIVERSITÉ DE GENÈVE

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

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

Parameters (input data); in this case:

- Capacity (P_i)
- Variable cost (c_i)
- Transmission capacity (T): no index i
- Load L_i

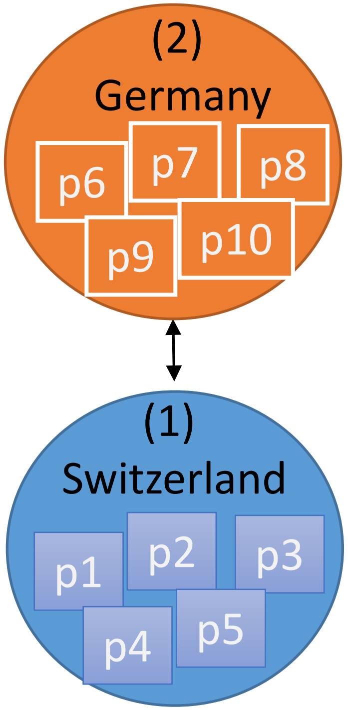


Objective function (what's being minimized/maximized):
In this case: minimize total cost

Constraints: Equations/inequalities limiting the variable choice

Structure of an optimization model: Sets

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



Example of node Sets and power plant sets

Structure of an optimization model: Sets (in the exercise)

UNIVERSITÉ DE GENÈVE

In the exercise (Pyomo)

- power plants and 4 seasons
- Sets are attributes (properties) of the model object m
- More simply: 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”:

`m.seasons * m.power_plants` (each power plant during each season)

- subset, e.g. wind and solar:

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

Structure of an optimization model: Parameters

UNIVERSITÉ DE GENÈVE

- Parameters are input data (costs, technical properties of assets, transmission capacities, etc)

In the exercise (Pyomo)

- costs, capacities, capacity factors, demand, efficiency, time slot (season) duration/weight…
- Just like sets, parameters are attributes (properties) of the model `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', '0_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, … etc.})
```

Structure of an optimization model: Variables

UNIVERSITÉ DE GENÈVE

- 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. and domains (real numbers, binaries, integer values)

In the exercise (Pyomo)

- 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

UNIVERSITÉ DE GENÈVE

- Energy balance constraints: for each time slot, the supply is greater or equal the demand

$p_{nuclear,t} + p_{gas,t} + p_{hydro,t} + p_{coal,t}$ $(+p_{storage,discharge,t}$

$- p_{storage,charge,t}\ldots) >= D_t$ for each time slot t

In the exercise (Pyomo)

- Function demand_constraint_equation is called for each season and returns an expression “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.season,
                                    rule=demand_constraint_equation)
```

Constraints: Formulation in Pyomo

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)`

required

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; analogously for power plants or "for each plant and each season": m.power_plants*m.seasons

Constraints: Other basic constraints – capacity I

Problem: We want to limit power production *for each power plant during each season*. Maximum power production for each plant is given by the installed capacity parameter `m.cap`

```
def capacity_constraint_equation(m, plant, season):
        return m.pwr[plant, season] <= m.cap[plant]

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

Construct a constraint for each given plant and season

No sum, each power plant/season individually

"for each power plant during each season"

Constraints: Other basic constraints – capacity II

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

=> Variable `m.cap_new`, exists only 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)
```

Variable only exists for new_power_plants

Parameter exists for all plants

Constraints: Other basic constraints

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

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

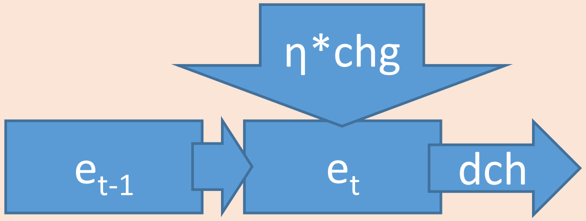
- The capacity factor follows from the resource availability (e.g. zero at night for solar, wind produces more power in winter as compared to summer)

Storage state-of-charge

For each storage plant and each time slot, the stored energy is equal the stored energy from the last time step plus charging minus discharging

$$e_{\mathrm{plant},t} = e_{\mathrm{plant},t-1} + w_t \cdot \eta p_{\mathrm{plant,chg},t} - w_t p_{\mathrm{plant,dch},t}$$

- Important: stored energy (MWh) calculated from charging/discharging power (MW) by multiplying with time slot weight w_t



Structure of an optimization model: Objective function

UNIVERSITÉ DE GENÈVE

- Single value
- Examples: Total cost (minimize), emissions (minimize), revenue/profit (maximize)
- Total cost: Fuel cost of all power plants, fixed and variable OPEX, CAPEX, (ramping costs, start-up costs) …
- e.g. fuel cost: vc[EUR/MWhfuel]/eff[MWel/MWhfuel]*p[MWel]*w[h/year]

Example here: Yearly cost

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

$$\mathrm{VC}_{\mathrm{OPEX,plant},t} = \mathrm{vc}_{\mathrm{OPEX}} \cdot p_{\mathrm{plant},t} \cdot w_t$$

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

$$\Rightarrow \mathrm{TC} = \sum_{\mathrm{plant}} \left\{ \sum_{t} \left[\mathrm{VC}_{\mathrm{fuel,plant},t} + \mathrm{VC}_{\mathrm{OPEX,plant},t} \right] \right\} + \sum_{\mathrm{plant}} \mathrm{FC}_{\mathrm{plant}}$$

Objective function: Formulation in Pyomo

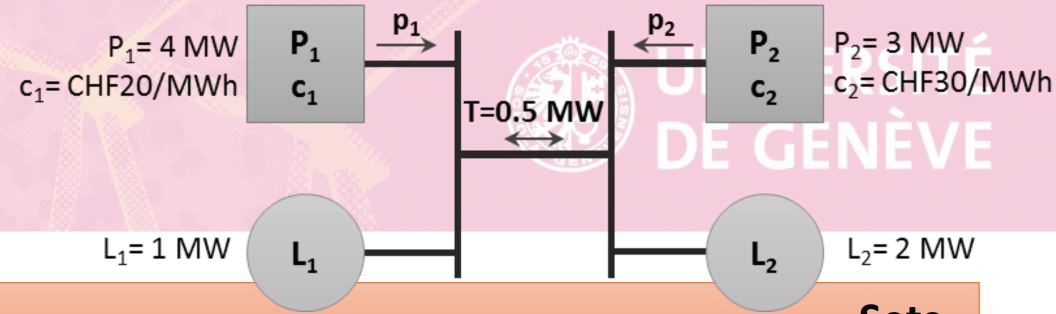
Note: double sum (all seasons, all power plants)

```
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 first 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.nregions, 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)
```

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

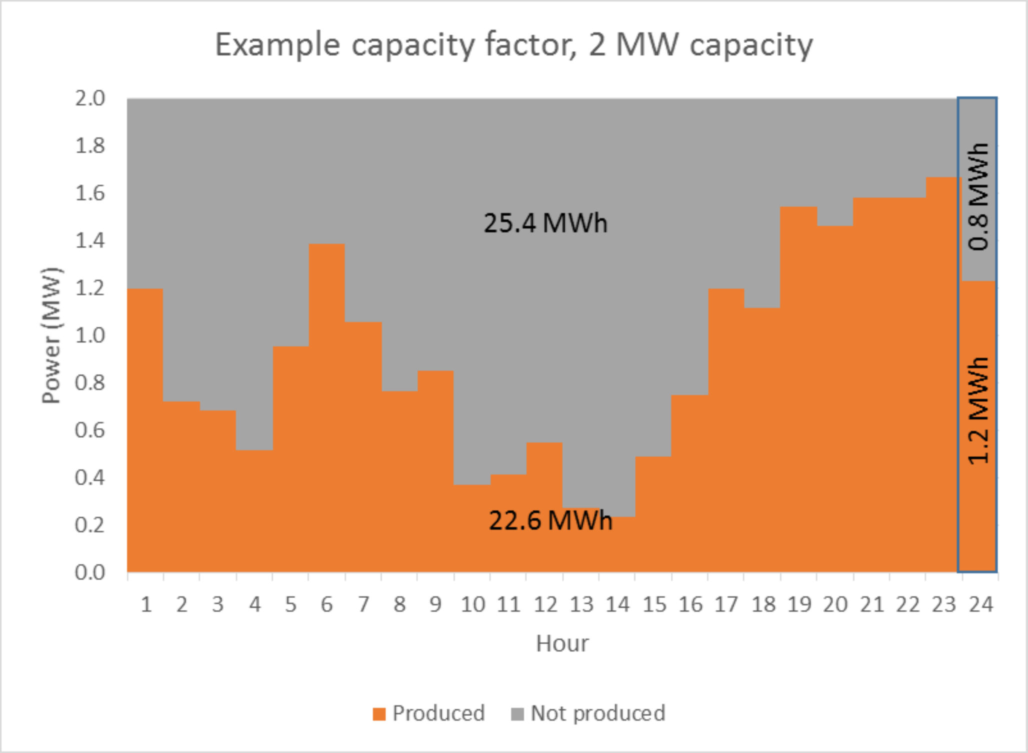
UNIVERSITÉ DE GENÈVE

Part III: Various important concepts

Some important concepts I: Capacity factor/Full load hours

UNIVERSITÉ DE GENÈVE

- **Capacity factor**: Ratio of produced energy and maximum energy produced at full nominal capacity
- **Full load hours (FLH)**: Capacity factor x 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)
- What causes electricity/heat/etc. generators to have a capacity factor < 1?



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

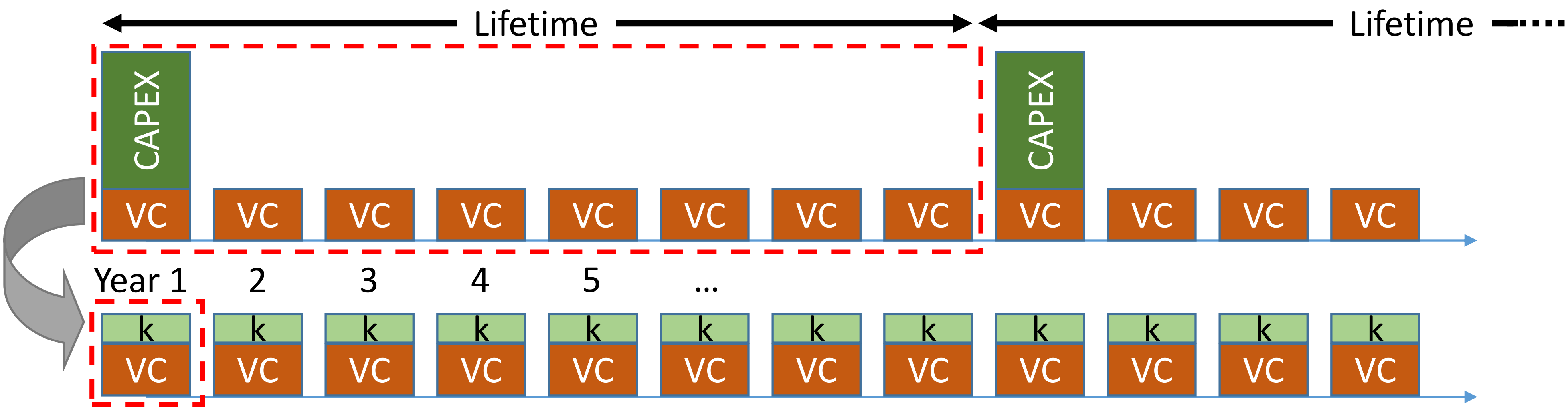
- *Daily cf:*

$$\frac{22.6\text{MWh}}{24\text{h} \cdot 2\text{MW}} = 47.2\%$$

- *Daily FLH*:
  $47.2\% \cdot 24\ \text{h/day} = 11.3\ \text{hours/day}$
- *Hourly cf (hour 24):*

$$\frac{1.2\text{MWh}}{1\text{h} \cdot 2\text{MW}} = 60\%$$

Some important concepts II: Annualized capital cost



$k = \text{CAPEX} \cdot \alpha$ and $\text{CAPEX} = \frac{k}{(1+r)^1} + \frac{k}{(1+r)^2} + \frac{k}{(1+r)^3} + \cdots$ with the annuity factor $\alpha = \frac{(1+r)^{\text{lifetime}} \cdot r}{(1+r)^{\text{lifetime}} - 1}$ and the discount rate r

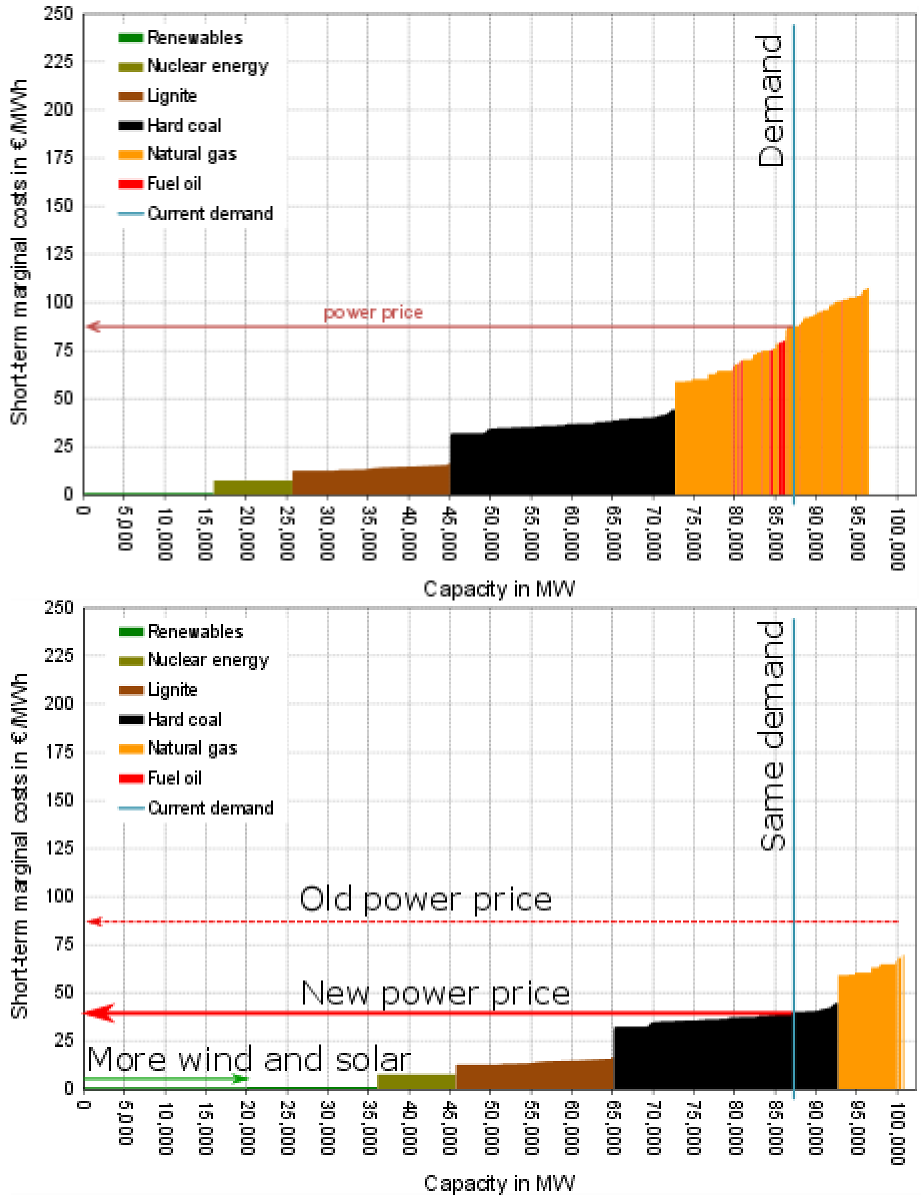
- If the operation during all years is the same we can calculate the **annualized investment cost** (CHF/MW → CHF/MW/year) and only consider a single year
- Then **the levelized CAPEX** is the annualized CAPEX divided by the full load hours: (CHF/MW/year) / (hours/year) → CHF/MWh
- Please revisit the week 6 assignment for further details

Some important concepts III: The «merit order effect»

UNIVERSITÉ DE GENÈVE

- 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

- More wind and solar power with zero-variable cost shift the supply curve to the right
- Therefore, 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)
- **Remember**: More wind and solar power → lower electricity prices during the wind and solar production hours → value erosion

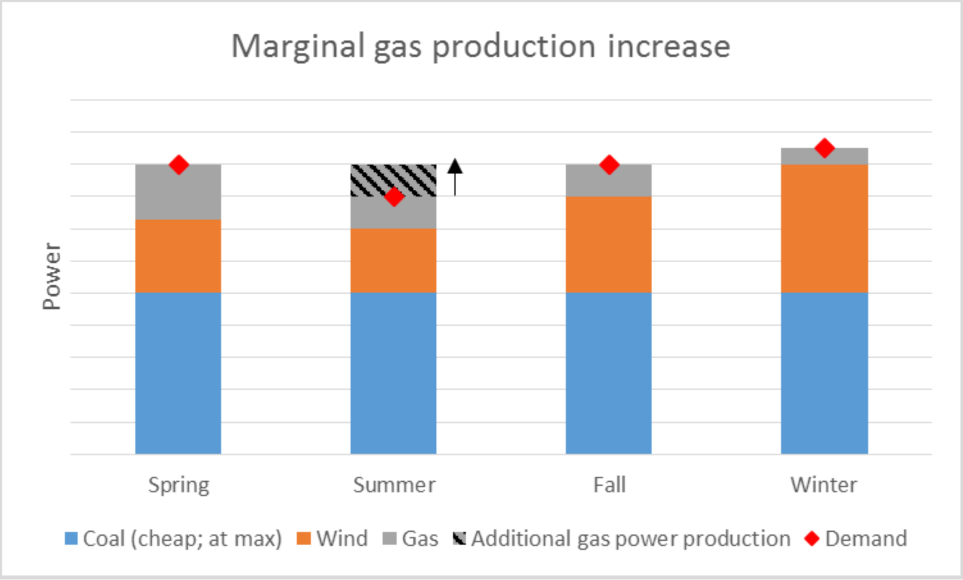


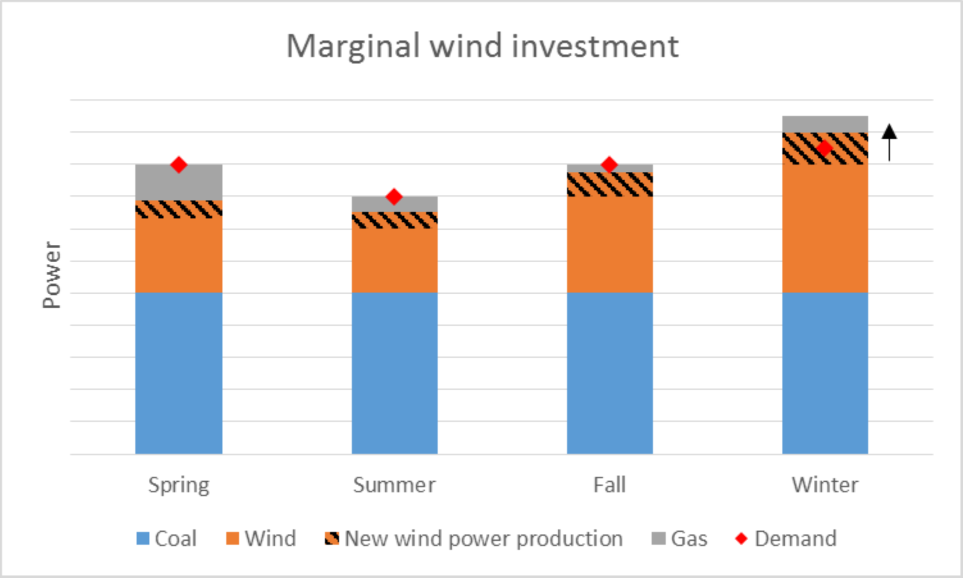
Adapted from http://dx.doi.org/10.1016/j.eneco.2014.04.020

Some important concepts IV: Shadow prices/Marginal generator

UNIVERSITÉ DE GENÈVE

- *<u>Shadow price</u> of a certain time slot in the model*: How much does the system price (objective function) change if the demand (power) is increased by one unit (CHF/MW)? → Electricity prices (CHF/MWh)
- Careful: Power (CHF/MW) vs Energy (CHF/MWh)





Simple case: 1 MW demand increase in summer leads to production increase from the cheapest available generator (**marginal generator**)

- How to calculate the shadow price in this case? Electricity price?

More complicated case: 1 MW demand increase in winter causes additional wind power installation and replacement of gas power production during the rest of the year.

- How to calculate the shadow price in this case? Electricity price?

Some important concepts V: Revenue

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

$$\text{revenue} = \mathrm{sum}_t \left(p_t[\mathrm{MW}] \cdot w[\mathrm{h}] \cdot \pi_t \left[\frac{\mathrm{CHF}}{\mathrm{MWh}}\right] \right)$$

When looking at the plant from the system's perspective, we can call this the "value".

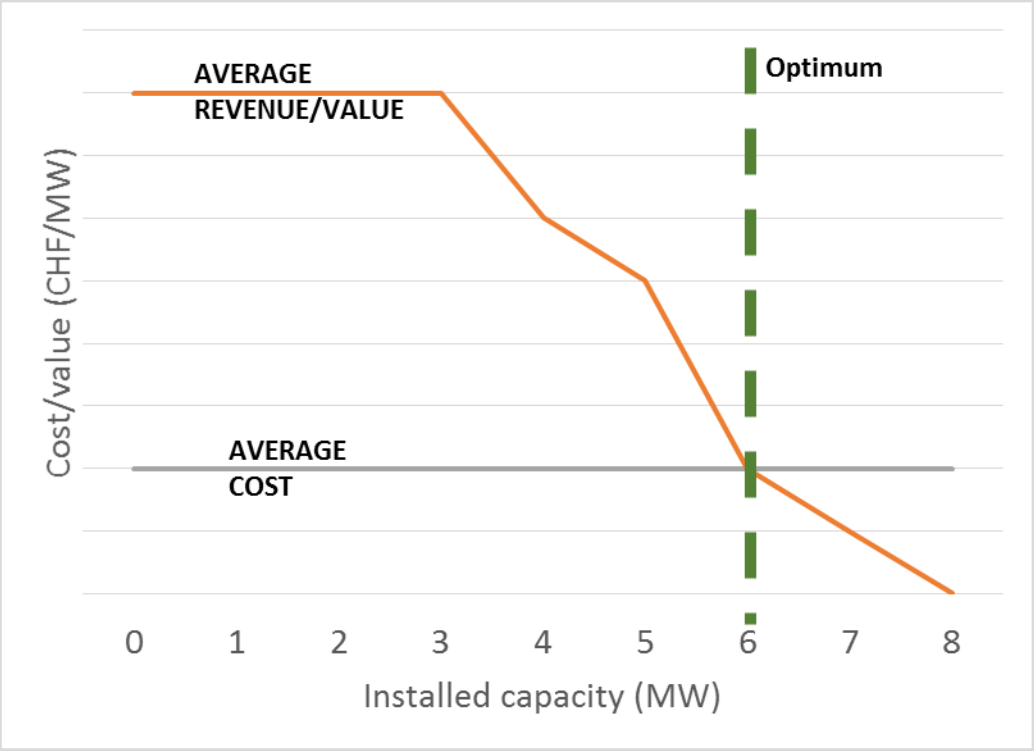
Some important concepts V: Cost = revenue/value

UNIVERSITÉ DE GENÈVE

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

Basic reasoning for the case of solar power:

- Higher capacity of the same solar resource **reduces the electricity prices** at noon
- The **average value/revenue of solar power is reduced** if more and more solar power is added to the system
- As long as not value = 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**

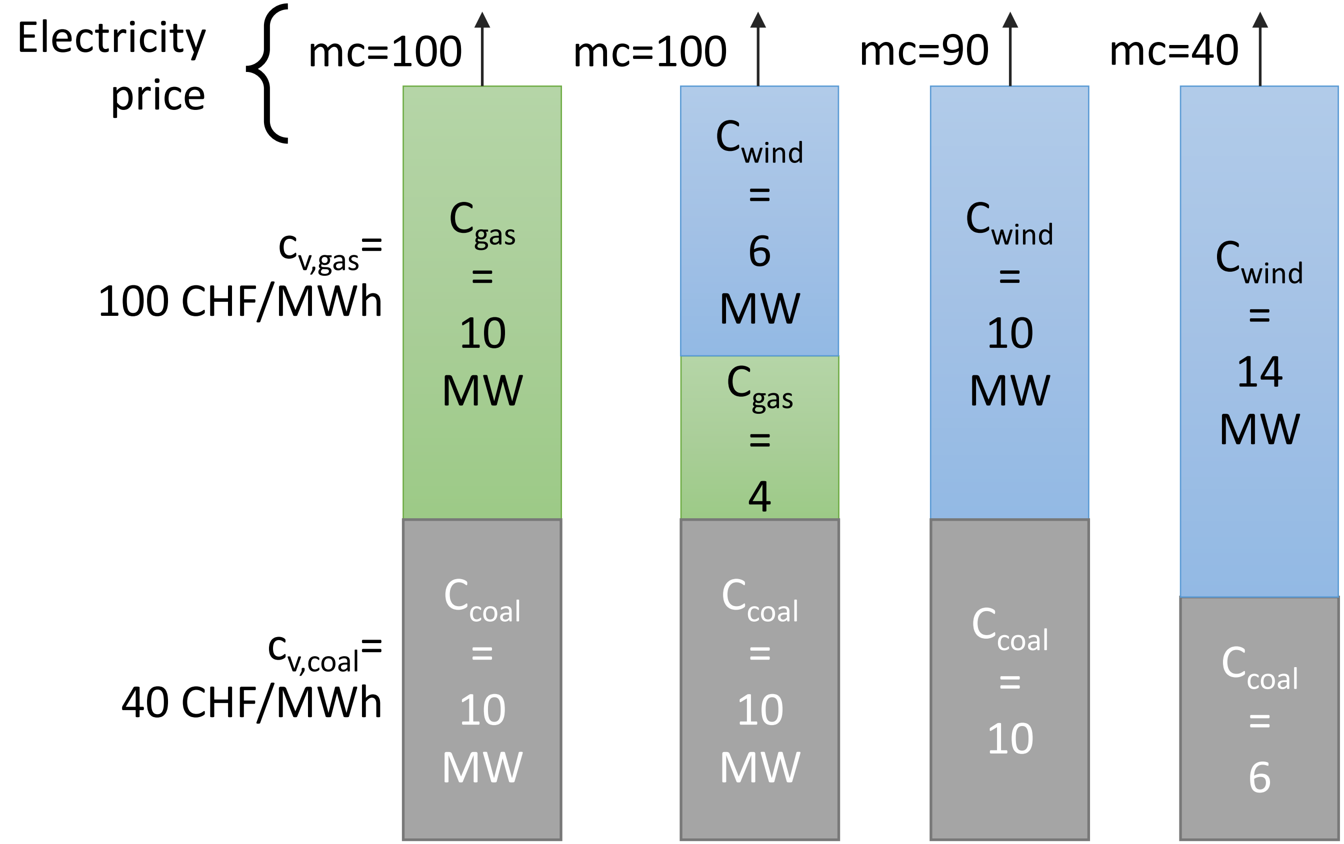


Some important concepts V: Cost = revenue/value, example

UNIVERSITÉ DE GENÈVE

- Single average hour
- Cheap wind power $LCOE_{wind}$=90CHF/MWh can replace gas power, but not 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*12
Wind cost (CHF/h)	-	90*6	90*10	90*12
Revenue - Value	**-**	**60**	**0**	**-600**

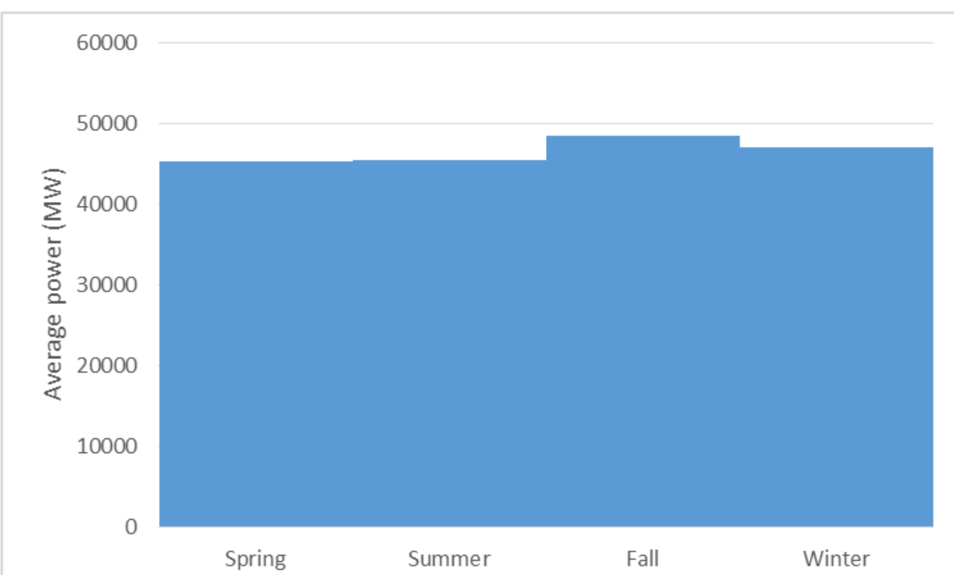
In the assignment: Show that this holds for more complex systems.

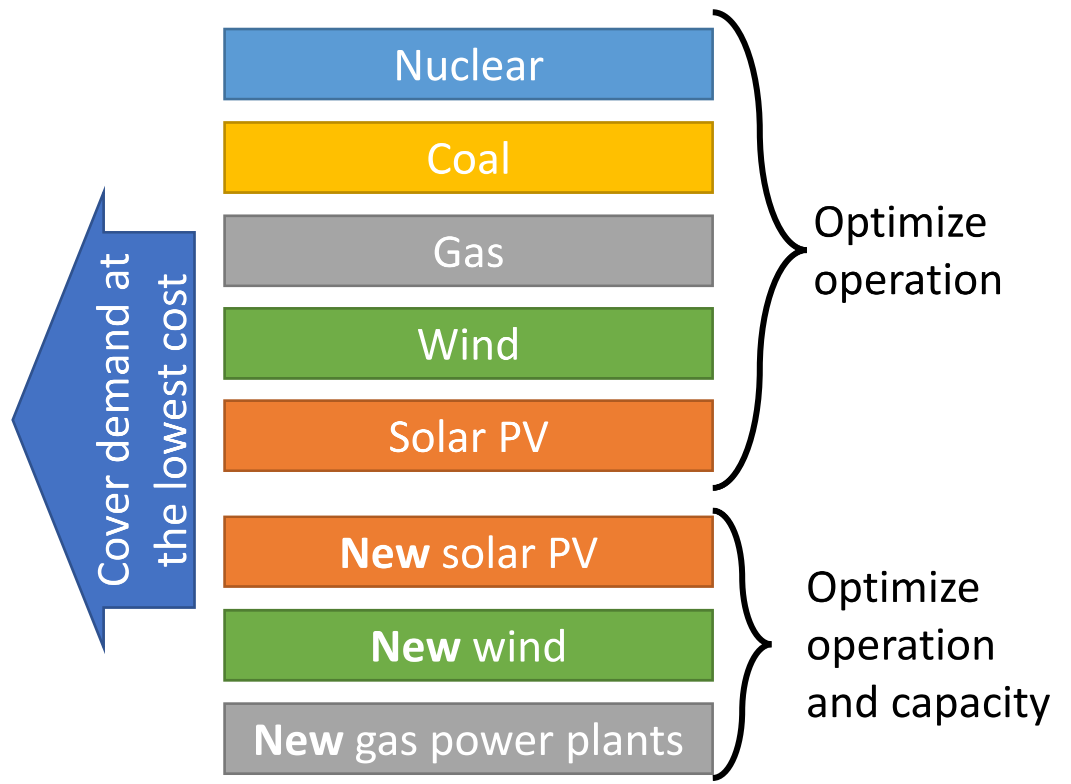
Part IV:
Notes on the assignment

Assignment

Purpose:

- Learn how to build a simple (but scalable) energy optimization model to represent the national electricity of a region
- How does the CO_2 price affect the optimal system composition?
- What happens to the wind revenue if we install more and more wind capacity?
- *BONUS*: Expand the model to assess the impact of energy storage on the wind value erosion





Assignment

UNIVERSITÉ DE GENÈVE

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

Assignment

UNIVERSITÉ DE GENÈVE

<u>Remarks on the exercise:</u>

- The exercise model is formulated in *Pyomo* (one of the Python optimization modules)
- We use Jupyter Notebooks (similar to *Week 7: Monte Carlo Simulation*)
- 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
- 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"; 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?

UNIVERSITÉ DE GENÈVE

Do not hesitate to reach out…

Martin C. Soini (martin.soini@unige.ch)
Arthur Rinaldi (arthur.rinaldi@unige.ch)

Institute for Environmental Sciences
University of Geneva
6th floor, office B604