Daily

Dienstag, 3. Dezember 2019 12:43

03.12.19:
- Necessary programs are installed on tum desktop
- Git tutorials are followed to learn better

03.12-09.12
- Git tutorials are completed

09.12
- Start with urbs docu https://urbs.readthedocs.io/en/latest/
- Working on different desktops are considered
- Environment is set up in personal computer

11.12
- Model description is done
- Model implementation is being read

13.01
- Back from holiday
- Catch up with docu+
- Read chapter 2-3 from Python book-
- Do mind map for thesis+

14.01
- Work on lab computers +
- Install gurobi ,for admin rights go to Leo the guy in front of the computer room +
- Try on the example model and first modify it (noted on future imp)
- It would be great if we can do an intertemporal model (noted on future imp)
- Rumne runned on single year example
- Run me modified
- Git repository cloned on lab computer
- Near_optimal branch is created for modifications
- Branch is pushed (results are at gitignore)

15.01
- Try to open project in pycharm to see objects better+
- Read chapter 4 of the book
- Try to modify scenarios, input etc+
- See how you can modify the objective function

16.01
- Near optimal mode is added to the code as an additional mode and a diffrenet python package is created for near optimal , scripts are copied with an added _nopt suffix
  features added without suffix
- Near optimal is integrrated now new conditional constrains should be added to model.py

17.01
- Model.py is read to understand pyomo set parameter etc. structure

```
ifm.mode['nopt']:
m.res_cost_restrict=pyomo.Constraint(
m,rule=res_cost_restrict_rule,
doc='costupperlimit=cost*cost_factor')
```

- ~~Is added to model.py~~
- ~~Cost restriction rule must be written in model.py~~
- ~~Cost restriction factor must be set somewhere as set or list [0.05 for example]~~

18.01
- Cost slack list is added inside input.py
- Cost rule adds costs together and then minimizes them:
  'Minimizing the value of the variable total system cost would give the most reasonable solution for the modelled energy system. '

```
#Ifthenearoptimalfeatureisactivatedsolvetheproblemfirstforcost
ifnear_optimal=='nearoptimal':
#createcostoptmodel
prob_co=create_model(data,dt,timesteps,'cost')
#solvecostmodelandreadoptimizedcost
optim=SolverFactory(Solver)#cplex,glpk,gurobi,...
optim=setup_solver(optim,logfile=log_filename)
result=optim.solve(prob_co,tee=True)
assertstr(result.solver.termination_condition)=='optimal'
opt_cost_sum=0
forkeyinprob_co.costs.keys():
opt_cost_sum+=prob_co.costs[key].value
forstfinprob_co.stf:
data['global_prop'].loc[(stf,'Cost_opt'),:]=opt_cost_sum
else:
pass
```

- This functionality is added to run function
- If near opt functionality is selected code solves the problem for cost objective first and then adds a optimum cost parameter to data
- Then continues the original sctipt

If you want to achieve getting the model with overwriting the objective: You can look mat pyomo core base objective.py
- ~~Next steps: add optimized cost as an attribute to model instance~~
- Implement in cost restrict rule
- Try ovarwriteing pyomo objective
- To overwrite obj I tracked down where objective is used in model creation

27.01
- I asked to LEO about changing objective without creating the model again
  He said it is hard to go into a pyomo model after creating it there are 2 possible ways to find out:
  Search from google if you can overwrite
  Look at the branch alternative scenarios A student try to run parallel models

- If you want to create model again
- ~~First try to do it with another objective for example co2~~

ISSU SOLVED : change model.py according to pull req according to: Bug fix modelhelper for intertemporal case
#261
- ~~CO2 objective is not working: Gives error at KeyError: 'Weight' while running stf_dist(stf,m) function at modelhelper.py,~~
- ~~Stf.dist function requires m.global_prob.loc[stf, 'Weight'] parameter which does not exist in single year example.~~
- ~~Stf_dist is called twice in model py one in res_global_CO2_budget_rule which is used by cost objective if model is intertemporal~~
- ~~Other one is co2_rule if the objective is co2 but here it is not only for intertemporal models~~

- After CO2 single year bug solved nopt model tried with 2 steps first solve for cost record the optimum cost solve for another objective (tried with CO2)
- ~~Now cost restriction will be added ++ (Now slack needs to be hard coded and it works only for first index of slack~~
- and minimize pv will be added as objective

- ~~Ok now you need to write a rule that sums the capacity of PV up. Find how algorithm calculates capacities +++~~

- The best approach to put capacity of a process as objective is to write a capacity rule. Which sums up the process capacity variable for all sites and ~~all years~~ last year . Capacity for a single year and single area has already been there
- There is a process capacity rule in model.py:

```
defdef_process_capacity_rule(m,stf,sit,pro):
ifm.mode['int']:
if(sit,pro,stf)inm.inst_pro_tuples:
if(sit,pro,min(m.stf))inm.pro_const_cap_dict:
cap_pro=m.process_dict['inst-cap'][(stf,sit,pro)]
else:
cap_pro=\
(sum(m.cap_pro_new[stf_built,sit,pro]
forstf_builtinm.stf
if(sit,pro,stf_built,stf)
inm.operational_pro_tuples)+
m.process_dict['inst-cap'][(min(m.stf),sit,pro)])
else:
cap_pro=sum(
m.cap_pro_new[stf_built,sit,pro]
forstf_builtinm.stf
if(sit,pro,stf_built,stf)inm.operational_pro_tuples)
else:
if(sit,pro,stf)inm.pro_const_cap_dict:
cap_pro=m.process_dict['inst-cap'][(stf,sit,pro)]
else:
```

```
cap_pro=(m.cap_pro_new[stf,sit,pro]+
m.process_dict['inst-cap'][(stf,sit,pro)])
returncap_pro
```

This is used to form an expression:

```
m.cap_pro=pyomo.Expression(
m.pro_tuples,
rule=def_process_capacity_rule,
doc='totalprocesscapacity')
```

- As this will be used as minimized variable cap_pro lets see how this kind of rule is used in cost minimization

Ok the logic works like that:

First if the objective is cost:

- Res_global_co2_limit_rule `'totalco2commodityoutput<=GlobalCO2limit')`
  ```
  return(co2_output_sum<=m.global_prop_dict['value']
  [stf,'CO2limit'])
  ```
- If intertemporal:
  - res_global_co2_budget_rule `totalco2commodityoutput<=global.propCO2budget`
  - res_global_cost_limit_rule, `'totalcosts<=Globalcostlimit`
- Objectivefunction : cost_rule `pyomo.summation(m.costs)`

First if the objective is CO2:

- res_global_cost_limit_rule, `totalcosts<=Globalcostlimit`
- If intertemporal:
  - res_global_cost_budget_rule, `totalcosts<=global.propCostbudget')`
  - res_global_co2_limit_rule, `totalco2commodityoutput<=GlobalCO2limit`
- Objectivefunction : co2_rule `pyomo.summation(m.costs)`

PV Objective:

- Cost restriction rule will be same as global cost limit rule but limiting value will be the optimized cost+ optimizedcost * slack +
- For intertemporal there will be a cost budget and co2 budget rule +
- Opjective: pv_capacity_rule +

28.01

- ~~?? Rule functions takes some arguments. Find how you give this arguments in constraint and objective definition:??~~
  - You should give only pyomo sets as input dont need to give m in the begining

  For example here m.tm ve m.com_tuples arguments olarak verilmis

  ```
  m.res_vertex = pyomo.Constraint(
      m.tm, m.com_tuples,
      rule=res_vertex_rule,
      doc='storage + transmission + process + source + buy - sell == demand')
  ```

  Com_tuples da böyle tanimli

  ```
  m.com_tuples=pyomo.Set(
  within=m.stf*m.sit*m.com*m.com_type,
  initialize=tuple(m.commodity_dict["price"].keys()),
  doc='Combinationsofdefinedcommodities,e.g.(2018,Mid,Elec,Demand)')
  ```

  Cünkü res_vertex_rule input olarak tm, stf, sit, com, com_type aliyor

  ```
  defres_vertex_rule(m,tm,stf,sit,com,com_type):
  #environmentalorsupimcommoditiesdon'thavethisconstraint(yet)
  ifcominm.com_env:
  returnpyomo.Constraint.Skip
  ifcominm.com_supim:
  returnpyomo.Constraint.Skip
  ```

  Ama concrete model disinda bir argument verebilir misin emin degilim: Artik eminim sadece pyomo set verebiliyosun!!!!

- Eger objective functionu overwrite edersen bu geliyor :

  WARNING: Implicitly replacing the Component attribute objective_function (type=<class 'pyomo.core.base.objective.SimpleObjective'>) on block urbs with a new Component (type=<class 'pyomo.core.base.objective.SimpleObjective'>). This is usually indicative of a modelling error. To avoid this warning, use block.del_component() and block.add_component().

- Code gives following error when obj is overwritten:

  ```
  Solved in 1300 iterations and 0.08 seconds
  Optimal objective  0.000000000e+00
  Warning: unscaled primal violation = 3.8147e-06 and residual = 3.8147e-06

  Process finished with exit code -1
  ```

- Now the code executes with PV objective, implemented code parts are:

  ```
  defcapacity_rule(m):
  capacity_sum=0
  ifm.obj.value=='pv':
  pro="Photovoltaics"
  #thisconditioncanbeutilized,whenothercapacitiyobjectivesareimplemented
  else:
  pro="Photovoltaics"
  forstfinm.stf:
  forsitinm.sit:
  capacity_sum+=def_process_capacity_rule(m,stf,sit,pro)
  return(capacity_sum)


  elifm.obj.value=='pv':
  defres_cost_restrict_rule(m,stf):
  try:
  cost_factor=m.cost_slack_list[0]
  cost_factor<1.0
  except:
  print('slackvalueisnotdefinedproperly.Slackvaluemustbeapositivenumberandsmallerthan1.0')
  ifm.global_prop_dict["value"][stf,'Cost_opt']>=0:
  return(pyomo.summation(m.costs)==(1+cost_factor)*m.global_prop_dict["value"][stf,'Cost_opt'])
  else:
  raiseValueError('optimizedcostisnotfoundinmodeldata')

  m.res_cost_restrict=pyomo.Constraint(
  m.stf,
  rule=res_cost_restrict_rule,
  doc='totalcosts<=Optimumcost*(1+e)')
  ifm.mode['int']:
  m.res_global_cost_budget=pyomo.Constraint(
  rule=res_global_cost_budget_rule,
  doc='totalcosts<=global.propCostbudget')
  m.res_global_co2_budget=pyomo.Constraint(
  rule=res_global_co2_budget_rule,
  doc='totalco2commodityoutput<=global.propCO2budget')

  m.objective_function=pyomo.Objective(
  rule=capacity_rule,
  sense=pyomo.minimize,
  doc='minimizePVcapacityemissions')
  ```

- capacity restriction rule is modified from <= to == and capacity minimization changed from summation of all years to only last year
- Yapilmasi gerekenler:
  - Minimuma maximumu da ekle
  - Farkli objectiveleri otomatize et
  - Create model yapmadan objectivei degistirmeye calis
  - Different slacklerde dene
  - Rapora ekle:

        - ~~Optimized cost~~
        - Capacities before cost optimized
        - ~~New cost~~
        - ~~Capacities after cost optimized~~
    - Plotlara ekle:
        - Capacity of pv with slack
        - Other capacities with slack
    - ~~Intertemporali dene~~

29.01

-Intertemporal mode is implemented capacity of pv decreased but it stayed equal between years
-????Cap pro ve cap_new icin cap_pro new we old total operational olanlari mi iceriyor?????

- So rapor icin önce get entities ile costs, cap_pro, cap_pro_new …. Gibi degerler data frame'e okunuyor. Get_constantsda bunlarin column ve index adlari düzenleniyor.
- Cap_pro sonuctaki process capsin total degeri
- Operational_pro_tuples buna yield ediyor:

```
instance.operational_pro_tuples.initialize
Out[6]:
[('Mid', 'Biomass plant', 2019, 2019),
 ('Mid', 'Biomass plant', 2019, 2024),
 ('Mid', 'Biomass plant', 2019, 2029),
 ('Mid', 'Biomass plant', 2019, 2034),
 ('Mid', 'Gas plant', 2019, 2019),
 ('Mid', 'Gas plant', 2019, 2024),
 ….
```

- For cap_pro for example onset names are ['stf', 'sit', 'pro']
- Cap_pro instanceda böyle tanimli :
    - (2019, 'Mid', 'Biomass plant') : cap_pro_new[2019,Mid,Biomass plant]
    - (2029, 'Mid', 'Photovoltaics') : cap_pro_new[2024,Mid,Photovoltaics] + cap_pro_new[2019,Mid,Photovoltaics] + cap_pro_new[2029,Mid,Photovoltaics]

- REPORT:
- A new sheet is added for optimized capacities it takes values from former page
- Optimum cost and total cost is added to cost page
- Do also optimum cost capacities

03.02

- Objective function is overwritten by del component. This way it is working now but must be checked: ??? What happened to the variables already calculated are they deleted or overwritten when problem is solved with a new objective???
- Now code deletes obj function after calculating cost optimum and storing the optimum cost and capacities of the process stated in real objective at cost optimum
- Then it defines cost restriction Constraint, co2 budget and limit constraints and defines a new objective function (only minimize now)
- Reports optimized capacities and capacities at cost optimum in one spread sheet
- Code is working for both single year and intertemporal
- Objective function.sense maximum ise -1 minimum ise 1

- Code can calculate min and max in the same loop and print the results on excel sheet
- Pushed as integrate min max together
- Nor near_optimal variable has depreciated since it is not needed

04.02

- ### adding other slack values
- Slacks are given as a list
- Code enters a condition if objective is not equal to cost or co2
- In this condition first calculates cost optimum
- Recods capacities that are going to be objected to minimization or maximization, and also records costs.

- This is done by 2 new functions
- Read capacity and read costs: which are defined in model helper

```
Def read_capacity(m,name):
objective_arg=m.cap_obj
cpro=get_entities(m,['cap_pro','cap_pro_new'])
ifnotcpro.empty:
cpro.index.names=['Stf','Site','Process']
cpro.columns=['Total','New']
cpro.sort_index(inplace=True)
optimized_cap=cpro['Total'][:,:,objective_arg]
optimized_cap=pd.concat([optimized_cap],keys=[objective_arg],names=['Objective'])
optimized_cap=optimized_cap.to_frame(name=name)
returnoptimized_cap
```

```
defread_costs(m,name):
objective_arg=m.cap_obj
costs=get_entity(instance,'costs')
total_cost=pd.Series(costs.sum(),index=['TotalCost'])
costs=costs.append([total_cost])
costs=pd.concat([costs],keys=[objective_arg],names=['Objective'])
costs=costs.to_frame(name=name)
returncosts
```

- Then model.objective_function is deleted with del_component()
- Cost restrict rule is defined
- From there a for loop for called for list of slack values
- In each loop first slack is defined to a model variable prob.cost factor
- Then it is checked if cost restrict constraint is already defined
- If it is defined it must be deleted first with del_component because it calls cost restrict rule which uses slack values
- Objective if first defined to minimize
- And problem solved
- Minimized capacities and new cost values are read by above read functions
- Model.objective_function again deleted with del_component for the purpose of preparing model instance for capacity maximization
- Maximizing objective function is defined
- Model solved
- maximized capacities and new cost values are read by above read functions
- Then all read capacities and costs are joined to 2 data frames for reporting purposes , data frames are saved to model variables
-
- Now report cost spread sheet reports all cost_types for each solved problem plus sum of each cost type for each column
- Capacities for each year and site of objective process for each solved problem

- Now sites subjected to minimization can be given in objective.

05.02

-Objective Tuples implemented

-Objectives now look like this

```
#exampleobjectives[('South','Photovoltaics'),('Photovoltaics'),('South','North','cost')]etc.
```

- this part is added to run scenario after data is read. Objective process and objected sites are read from tuple to seperate variables.
- Site is read in a list it can be multiple
- If only process is given it means all sites are objected.
- In that case since tuple has only 1 element. Python reads it as a list. That is why it is read seperately.
- From data site names are found and witten to sites list.

```
ifisinstance(objective[0],tuple):
objective_pro=objective[0][-1]
objective_sites=list(objective[0][:-1])
else:
objective_pro=objective[0]
objective_sites=[]
site_dict=data['site'].to_dict()
forkeyinsite_dict['area']:
objective_sites.append(key[1])
```

-objective process and list of objected sites are save in different model variables
-objected sites are added to report and plot names

????????????-Capacity rule is changed to sum up only objected sites

```
defcapacity_rule(m):
capacity_sum=0
pro=m.cap_obj
sites=m.cap_sites
iflen(sites)==0:
sites=m.sit
stf=max(m.stf)
#forstfinm.stf:
forsitinsites:
capacity_sum+=def_process_capacity_rule(m,stf,sit,pro)
```

```
return(capacity_sum)
```

-plotting is being handeled next.

- As first step I read how existing plots are build in the code since my plotting skills in matplotlib are a bit better than plotting skills of paint itself
- Gri cizgi unshifted demand side management timeseries original adli var. Icinde tutuluyor
- Siyah cizgi de demand timeseries ama shifted yani hourstep[1:]

- For plotting only `prob.near_optimal_capacities` are needed
- This is a multi index data frame
- How to handle it
- This is complete df

```
                                        Minimum Cost  ...     Min-0.01
Objective_Process Objective_Sites       Stf  Site           ...
Photovoltaics     ['Mid', 'North', 'South'] 2020 Mid     2189.065471  ...     0.000000
                                        North  39434.420377  ...   1672.116816
                                        South  10936.391977  ...  10907.783630
```

- `prob.near_optimal_capacities.sum(level=['Objective_Process', 'Objective_Sites', 'Stf' ])`
- This sums up capacities at mid south north
- Also possible to create a stacked plot

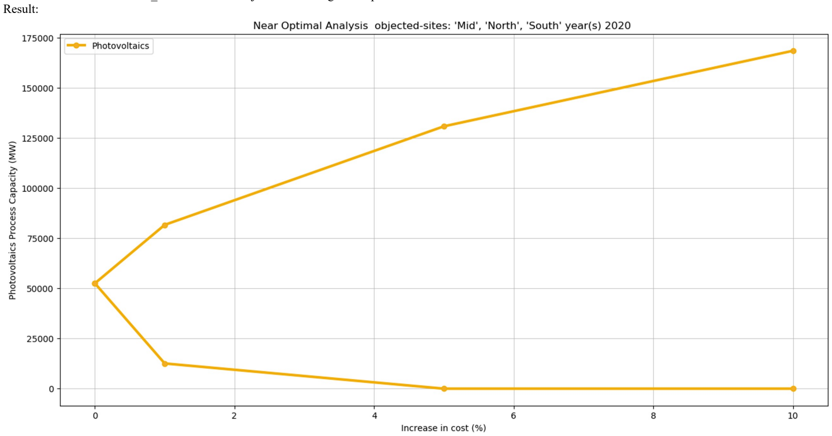
- How to reach values of different indexes

```
summed_cap.loc['Photovoltaics']['Minimum Cost']
Out[8]:
Objective_Sites              Stf
['Mid', 'North', 'South']  2020     52559.877824

summed_cap.loc['Photovoltaics',"['Mid', 'North', 'South']"]['Minimum Cost']
Out[10]:
Stf
2020     52559.877824
Name: Minimum Cost, dtype: float64
```

03.02

- A seperate plot function is written inside plot_nopt
- This function is designed for single year
- In this function:
  - Three arrays are formed: slack, cap_max, cap_min
  - `prob.near_optimal_capacities.sum(level=['Objective_Process','Objective_Sites','Stf']).loc[prob.cap_obj,str(prob.cap_sites),2020]['Min-{}'.format(slack)])`
  - Max and min capacities fior each slack is saved in those arrays and capacity arrays are ploted agains slack*100
  - Figure is saved to respective result folder
  - This function does not return anytrhing
- Then function is called in run_scenarios function just after exiting for loop that handles different slacks
- Result:



11.02

- Todays plan to implement multiple objectives
- Birden fazla objective olacak, bunlarin birbirinden farkli sitelari da olabilir
- Bu objectiveleri ayni excelle mi report edecegiz yoksa farkli mi
- Plotlari üst üste mi yapacagiz ayri mi
- Her objective icin model bir daha create ediliyor. Bunun sebebi costtan caapcity okumak gerekmesi bunu yerine costu bir üst loopa cikartip capacityleri önden okumak mantikli olabilir

12.02

- Code is reverted to the checkpoint before multiple objected and re designed for new goals.

- Code is modified to read multiple objectives and for each objective process objected sites can be assigned flexibly.

- Objective looks like this: `[('South','Photovoltaics'),('Windpark')]`

- Objective (tuple, list) is unpacked to a dictionary with processes as keys and corresponding sites as key values. Function, that performs unpacking is given below:

```
def unpack_obj(objective, data):
    objective_dict = {}
    for k, item in enumerate(objective):
        if isinstance(objective[k], tuple):
            objective_pro = objective[k][-1]
            objective_sites = list(objective[k][:-1])
            objective_dict[objective_pro] = objective_sites
        else:
            objective_pro = objective[k]
            objective_sites = []
            site_dict = data['site'].to_dict()
            for key in site_dict['area']:
                objective_sites.append(key[1])
            objective_dict[objective_pro] = objective_sites
    return objective_dict
```

- Objective sites can be assigned flexibly also for cost and CO2 objectives, however for those two specific objectives, it is assumed that objective list has only one tuple element. As objective= [('South','cost')] or =['cost'] or ['South','Mid','CO2'] etc.
- Here is psydocode for run_scenarios

Begin run_scenarios

Data= read_input
Objective_dict = unpack_obj

If objective is cost or co2:
Create model for given process
minimize/maximize total cost/CO2 for given sites
Plot figures
Write report

Else:
Create model for objective: cost

Minimize cost

Save minimum cost to a model variable

Read process capacities and costs from model solution to data frames

Define cost restrict rule

For each slack value defined:
  Delete model objective

  delete cost restrict constraint if already defined

  Define cost restrict constraint

  Define minimization objective

  Solve for minimization objective

  Read process capacities and costs from solution to data frames

  Delete model objective

  Define max objective

  Solve for max objective

  Read process capacities and costs from solution to data frames

  Merge process capacity data frames from minimization and maximization to minimized cost process capacity data frame

  Merge cost data frames from minimization and maximization to minimized cost /cost data frame

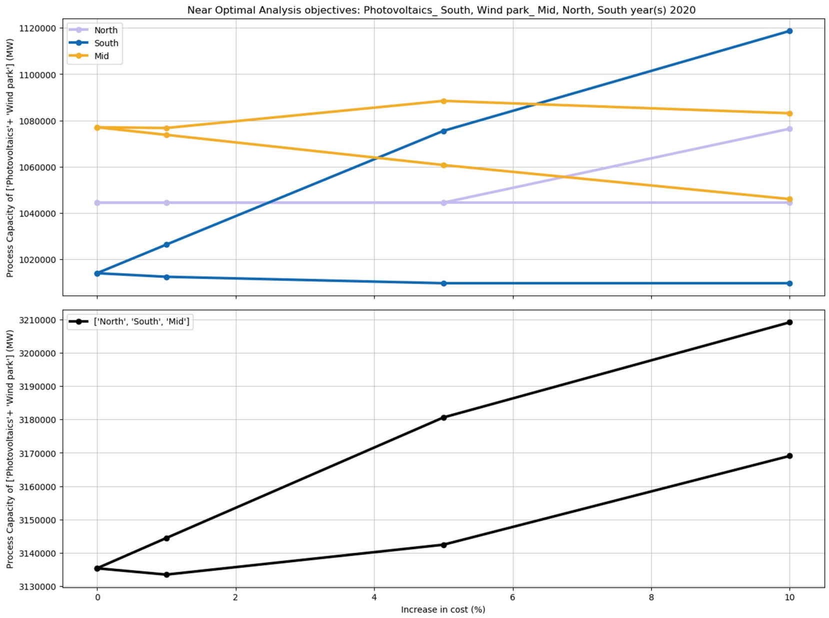
Report merged dataframes#
Plot data

17.02

- In order to extend objected site flexibility to cost and co2 objectives cost rule and CO2 rule are changed
- First Cost rule:
  - Eger objective cost ve ya co2 degilse costu tüm çözümler icin okuyup tek bir dataframee kaydedip reportsa onu yolluyorum
  - Costlar def_cost_rule'da hesaplaniyor
  - m.def_costs=pyomo.Constraint(
  - m.cost_type,
  - rule=def_costs_rule,
  - doc='maincostfunctionbycosttype')
  -
  - Costs are variables in cost type form
  - m.costs=pyomo.Var(
  - m.cost_type,
  - within=pyomo.Reals,
  - doc='Costsbytype(EUR/a)')
  -
  - Def_cost_rule m.costsun icini dolduruyor
  - Def_costs constraint def_costs_rule u cagiriyor
  - Solver bu constainti çözerek olusacak costlari hesapliyor ve bu costlar
    - For p in m.pro_tuplesla hesaplaniyor
    - m.pro_tuples=pyomo.Set(
    - within=m.stf*m.sit*m.pro,
    - initialize=tuple(m.process_dict["inv-cost"].keys()),
    - doc='Combinationsofpossibleprocesses,e.g.(2018,North,Coalplant)')
    -
  - Calculation of costs are modified also for site selection. This function is activated only when cost is the optimization objective
  - Transmission_costs and storage_costs functions are modified also for site selection. This function is activated only when cost is the optimization objective
    - `if t[1] in m.objective_dict['cost'] or t[2] in m.objective_dict['cost']` for transmission_costs both site in and site out are checked for selected sites
    - `For s in m.sto_tuples if s[1] in m.objective_dict['cost']`
  - Revenue_costs function and purchase costs are also modified similarly
  - CO2 rule also modified it was straightforward

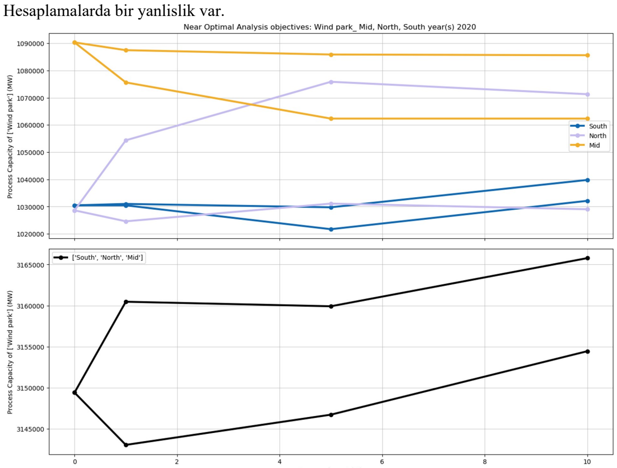
- For near optimal objective pre Cost minimization all sites should be taken regardless of choosen sites that is why original fuctions are held for this optimization

- Plotting for single year example is done
- Cummulative capacities for objected processes are shown for different regions in top plot and sum of those three graphs are given in the plot below



18.02

- Hesaplamalarda bir yanlislik var.

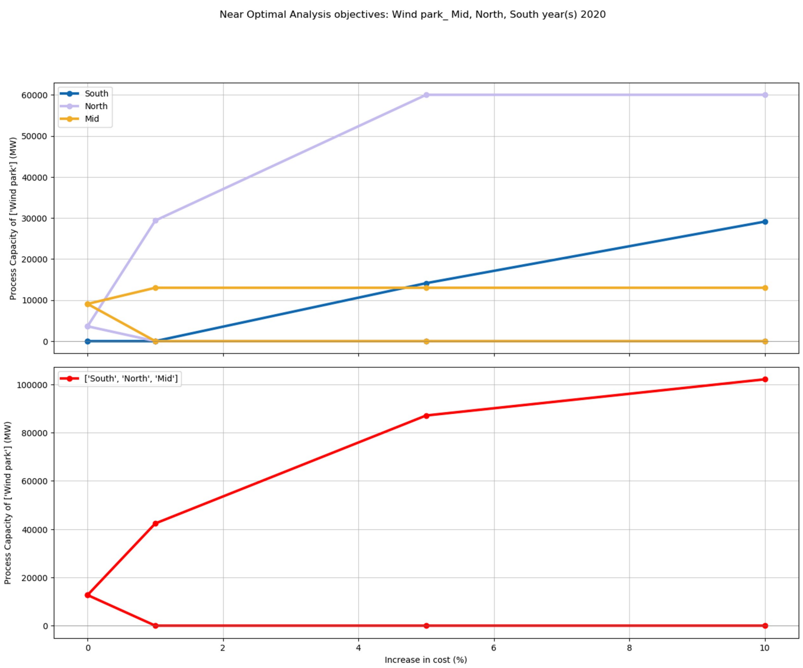


- Bu optimizasyonda windparklarin tamaminin optimizasyonu yapiliyor minimumun optimum costtaki kapasitenin altinda cikmasi lazim. Maximumun da slack arttikca artarak gitmesi lazim
- Bu ya graph ciziminden ya da hesaplamadan kaynakli bir hata

- Code is followed from top to bottom manually and psuedocode is read to catch any invisible bugs happening

19.02

- Bug is fixed



Bug issue is solved, selection of summed capacities durig plotting was false

-Reporting for single year is next
Put objective as index to capacity spreadsheet

- If objective is cost or co2 report is in the old format
- If objective is near optimal report has 2 spreadsheets one for costs report with results for max min objectives different slacks and oen for capacities

26.02

- In the meantime I created stack bar and area plots
- Now I am trying to run germany file but storage does not work
- There is a problem in add storage which does not appear on original file
- Problem comes from the ep ration tuples. In original file ep ratios are empty however in germany file there is a battery storage model so ep ratio of it is present
- In `def_storage_energy_power_ratio_rule` red marked stf indexes were missing then problem was solved

```
#storagetuplesforstorageswithgivenenergytopowerratio
m.sto_ep_ratio_tuples=pyomo.Set(
within=m.stf*m.sit*m.sto*m.com,
initialize=tuple(m.sto_ep_ratio_dict.keys()),
doc='storageswithgivenenergytopowerratio')
```

sto_ep_ratio_tuples : storages with given energy to power ratio
Dim=0, Dimen=4, Size=0, Domain=sto_ep_ratio_tuples_domain, Ordered=False, Bounds=(None, None)
[]

m.sto_ep_ratio_dict.keys()
: dict_keys([])

```
m.def_storage_energy_power_ratio=pyomo.Constraint(
m.sto_ep_ratio_tuples,
rule=def_storage_energy_power_ratio_rule,
doc='storagecapacity=storagepower*storageE2Pratio')
```

```
defdef_storage_energy_power_ratio_rule(m,stf,sit,sto,com):
return(m.cap_sto_c[stf,sit,sto,com]==m.cap_sto_p[stf,sit,sto,com]*
m.storage_dict['ep-ratio'][(stf,sit,sto,com)])
```

- Germany file has been run. Plot colors were adjusted
- Stackplot legends are organised so that processes with 0 capacity wont appear in legend and no legend will be repeated.
- Legend colors is sorted , to make them in the same order as in plot.

02.03:

- Bugün input file'in raporundan gerekli bilgileri ve sourcelari toplamaca ve teze genel bir göz atmaca
- Sonra zaman kalirsa intertemporalde run etmece

Related data from input thesis

- Außerdem werden für jeweilige Variante des Strombereitstellungssystems die kostenoptimalen Anforderungen festgelegt, sodass eine CO2-Einsparung von 95% gegenüber 1990 mit Variation der Nachfrage geschaffen werden kann.
- ** CO2 is reduced by 95% from 1990 levels

- Die Kohle- und Ölkraftwerke sind zuerst ausgeschlossen, da diese einen höheren durchschnittlichen CO2-Ausstoß aufweisen **is that realistic

- Der CO2-Austoß der GuD-kraftwerke wurde für alle Modellrechnungen auf 201,6 t/GWhth (1) festgesetzt.
- **CCGT co2 output rati 201.6 ton/GWhth

- Annahmen bezüglich der Inputparameter getroffen werden: (Assumptions)
    - Festsetzung einer Steuer für CO2
    - Festsetzung einer CO2-Grenze gegenüber dem Jahr 1990
    - Bestimmung ganzjähriger stündlicher Zeitreihe für den Strombedarf

- Übertrifft in einem Zeitpunkt die Erzeugung den Verbrauch, so muss negative Ausgleichsleistung bereitgestellt werden. Für den Fall dass die Erzeugung unter dem Verbrauch liegt, muss positive Ausgleichsleistung bereitgestellt werden. (curtailment and slack)

- Für die Investitions- und Fixkosten der Elektrolyseuranlagen bzw. der geothermischen Kraftwerke wurde eine billige bzw. teure Version angenommen, da derer zukünftige Kostenentwicklung nicht so klar war.
- ** Our input material will use billige versionen because they are the assumptions for 2050

-

- Im Laufe der Recherche soll die Frage beantwortet werden, wie lässt sich das Stromsystem von der CO2-Besteuerung statt der CO2-Begrenzung beeinflussen.

- Die Schritte zur Kostenoptimale Entwicklung eines Modellszenarios sind:
    - CO2-Steuer auf Null setzen
    - 95% Minderung der CO2-Emissionen als Nebenbedingung (from which years level:1990)
    - Rechnung des dualen Schattenpreises co_preis (What?)
    - Schrittweise Erhöhung der CO2-Steuer jeweils um 10% von 0 bis co_preis (grow up by 10% from 0 to co_preis)
    - Optimierung mit CO2-Besteuerung statt CO2-Begrenzung

** Shattenpreis (shadow price):

A **shadow price** is commonly referred to as a monetary value assigned to currently unknowable or difficult-to-calculate costs. It is based on the willingness to pay principle - in the absence of market prices, the most accurate measure of the value of a good or service is what people are willing to give up in order to get it. Shadow pricing is often calculated on certain assumptions and premises. As a result, it is subjective and somewhat imprecise and inaccurate.

Aus <https://en.wikipedia.org/wiki/Shadow_price>

** Explanation of how urbs handles CO2 balance and environmental comodities are explained in URBS page

- Die Methodik für jedes Modellszenario:
  - Jede Systemvariante entwickelt sich unter Annahme einer Minderung der CO2-Emissionen um 95% gegenüber 1990.
  - Die CO2-Steuer beläuft sich immer auf Null.
  - Optimierung jeweiliger Systemvariante mit schrittweiser Erhöhung der Stromnachfrage jeweils um 10% bis zur Verdoppelung der Stromnachfrage von 2017.

- Als Basisstromnachfrage für die Modellrechnungen wurde die Stromnachfrage von 2017 verwendet, die bei 508 Terawattstunden liegt. Die stündlichen Zeitreihen wurden aus der Bundesnetzagentur abgeleitet (18).

- Die modellierten Zeitreihen mit den Kapazitätsfaktoren von PV, Onshore Wind und Offshore Wind stammen aus der Webseite renewables.ninja (19). Für die einknotigen Modelle wurden direkt die geforschten Zeitreihen eingesetzt.
-
- Die Stromerzeugungskapazität besteht aus GuD-Kraftwerken, PV, Onshore-Wind und Offshore-Wind. Die GuD-Kraftwerke stellen zusammen mit den Pumpspeichern (kein weiterer Ausbau erlaubt) die einzigen Möglichkeiten zur Flexibilisierung des Stromnetzes dar (Back-up Kapazität). Außerdem wurde der Stromaustausch mit Ausland ausgeschlossen. (is our excel input for the reference scenarion)
** Stromerzeugungskapazität und Speichersysteme
-Gas and steam turbine (CCGT) power plants
- Onshore wind turbines
- Offshore wind turbines
- photovoltaic systems
- Biomass power plants
- biogas power plants
- Geothermal power plants
- Pump storage
- Battery storage
- H2 Storage
- Thermal storage

- Die Methodik für jedes Modellszenario:
  - Die CO2-Reduktion wurde auf 95% gegenüber 1990 festgesetzt,
  - während keine CO2-Besteuerung angenommen wurde.
  - Die Stromnachfrage des Systems wurde mit den Zeitreihen von 2017 modelliert.

- Allerdings war eine Deckung des Strombedarfes unter diesen Annahmen nicht möglich. In einem nächsten Schritt wurden sowohl die CO2-Besteuerung als auch die CO2-Begrenzung ausgelassen.

03.02

- Rapora bakmaya devam

-

Solar- und Windanlagen			
	PV	Onshore Wind	Offshore Wind
Investionskosten in €/MW	436.000 (3)	865.000 (3)	1.285.000 (3)
Fixkosten in €/MW	8,720 (3)	17.300 (3)	25.700 (3)
Varable Kosten in €/MWh	0 (4)	5,0 (4)	5,0 (4)
Kapitalzinssatz in %	0,021 (4)	0,025 (4)	0,048 (4)
Abschreibungsdauer in Jahren	25 (4)	25 (4)	25 (4)
Installierte Leistung in MW (Norden)	20.460 (5)	40.305 (5)	5.427 (5)
Installierte Leistung in MW (Süden)	21.879 (5)	9.986 (5)	0 (5)
Ausbaupotenzial in MW (Norden)	146.319 (6)	110.318 (7)	54.000 (7)
Ausbaupotenzial in MW (Süden)	78.084 (6)	87.683 (7)	0 (7)

- ** Bu datalar excelldeki datalar. Kapitalzinssatz : kapital interest rate . Ausbau potatiallerin toplami cap-up da verilmis.

Erdgas und Biomasse			
	GuD-Kraftwerk	Biomasse-kraftwerk	Biogas-Kraftwerk
Brennstoffkosten in €/MWh_{th}	34 (4)	28 (8)	28 (8)
Investionskosten in €/MW_{el}	800.000 (3)	2.000.000 (8)	3.000.000 (8)
Fixkosten in €/MW_{el}	30.000 (3)	60.000 (8)	44,400 (8)
Variable Kosten in €/MWh_{el}	4,0 (4)	1,0 (8)	1,0 (8)
Kapitalzinssatz %	0,052 (4)	0.027 (8)	0.027 (8)
Abschreibungsdauer in Jahren	33 (1)	25 (1)	33 (1)
Installierte Leistung in MW (Norden)	5.654 (9)	-	-
Installierte Leistung in MW (Süden)	2.346 (9)	-	-
Verfürgbare Primärenergie in MWh (Norden)	-	128.000.000 (10)	38.726.000 (6)
Verfürgbare Primärenergie in MWh (Süden)	-	134.000.000 (10)	33.507.000 (6)
Effizienz (%)	64% (1)	37% (8)	37% (8)
CO2-Austoß in t/GWh_{th}	201,6 (1)	0	0

** investment variable fix costs gud neu da, dummy gas icin inv 1 fix costs 0 variable o depreciation 0 ve bunlarin ikisinin birden capacity up i 0
** installed capacity dummy gas
** bu durumda sistem gud neu yerine dummy gas build etmez mi ???
**biogass kraftwerk investment costs are 110000 in excell it is not same here same for feste biomass. OK investment costs are investment cost in the paper times efficiency what is the purpose of that?? Also fix costs and inv costs from biomass is calculated same way but fix costs of biogas is not ?

** bohrung ve geothermische billig alinmis yani 2050 nin verileri

** okay elektrilizörde de söyle bir durum var. Electtrylisör elec alip h2 veriyor dummy h2 h2 alip gud veriyor gud da
Gud comm alip elec veriyor
**?????????? Should I show dummy H2 dummy gas gud and elekrolyseur as CCGT on graphs??? Similarly Geothermal well and geothermal pp

** Should I show curtailment on graphs

```
## Elect---->elektrolyseur-----> dummy h2 ------>
                                                  Gud----> Elec
##Gas---->dummy gas-------------------------------->
```

- REFERENZSCENARIO: In einem ersten Schritt sollte das Referenzsszenario unter diesen Annahmen optimiert werden: Die CO2-Reduktion wurde auf 95% gegenüber 1990 festgesetzt, während keine CO2-Besteuerung angenommen wurde. Die Stromnachfrage des Systems wurde mit den Zeitreihen von 2017 modelliert. (Bu bizdeki excel + co2 limited)

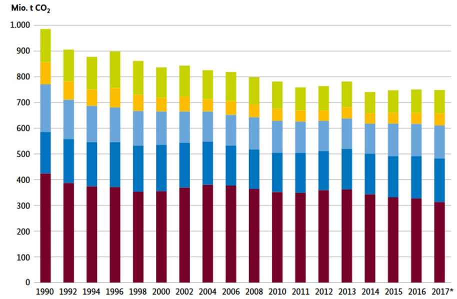
**
Germany aims to cut greenhouse gas emissions (GHG) by 40 percent by 2020, by 55 percent by 2030 and up to 95 percent in 2050, compared to 1990 levels. The share of renewables in gross final energy consumption is to rise to 60 percent by 2050. Renewables are to make up a minimum of 80 percent of the country's gross power consumption by the middle of the century.

Aus <https://www.cleanenergywire.org/factsheets/germanys-greenhouse-gas-emissions-and-climate-targets> (roadmap var compare edebiliriz yil limiti ve tek budgetla mi daha ucuz)

3.4 Verbrennungsbedingte CO_2-Emissionen im Energiebereich

Im Jahr 2017 lagen die CO_2-Emissionen aus der Verbrennung von Brennstoffen im Energiebereich bei 749 Millionen Tonnen CO_2. Damit wurden rund 24 Prozent weniger CO_2 als im Jahr 1990 freigesetzt.

Abbildung 10: Entwicklung der verbrennungsbedingten CO_2-Emissionen 1990 – 2017



	1990	1992	1994	1996	1998	2000	2002	2004	2006	2008	2010	2011	2012	2013	2014	2015	2016	2017*
Haushalte	129	123	128	142	132	118	121	113	113	107	106	90	94	100	82	87	90	90
Gewerbe, Handel, Dienstleistungen**	86	72	63	76	63	54	58	48	54	49	47	43	41	44	41	43	44	48
Industrie***	185	153	141	135	135	129	121	117	119	126	124	122	117	118	117	126	125	129
Verkehr	162	171	172	175	179	181	175	168	156	153	153	155	153	158	158	161	165	169
Energiewirtschaft	424	387	374	371	353	355	369	380	377	364	352	349	359	362	343	331	327	313
Summe	**986**	**907**	**878**	**900**	**863**	**836**	**844**	**827**	**819**	**799**	**782**	**758**	**764**	**781**	**742**	**747**	**752**	**749**

https://www.bmwi.de/Redaktion/DE/Publikationen/Energie/energieeffizienz-in-zahlen-2018.pdf?__blob=publicationFile&v=16

- In the excell CO2 limit is set to 999.999.999 which is bigger than 1990 emmitions. It is practically unset

- EXCEL Coppied and modified
  CO2 limit set to 0.05% of 1990 levels
- Names of processes commodities are changed to english also deutschland is changed with germany
- I am not sure if the solution is correct how can I validate my model and results?????
- Capacity up lari kaldirmam birsey degistirmedi cünkü zaten maximumlar capacity upa kadar cikmiyor
- Ama isin ilginci co2 limit de birsey degistirmedi
- Should I also consider only offshore, onshore, pv, biomass,CCGT, Fuelcell,Geothermal powerplant, biogas when I am evaluating the capacity stackplots
- Model didnt build any biomass biogas geothermal and fuelcell dolayisi ile bu bana sadece referans modeli yapiyormus gibi geldi

- Die Stromerzeugungskapazität besteht aus GuD-Kraftwerken, PV, Onshore-Wind und Offshore-Wind. Die GuD-Kraftwerke stellen zusammen mit den Pumpspeichern
- ** neden sadece pumped storage ve neden curtailment yok (üretmedigi icin mi)
- Bis zu 60% des Strombedarfs wird durch die fluktuierenden erneuerbaren Energien (PV, Onshore- und Offshore-Wind) gedeckt. Die gesamte Stromproduktion von PV, Onshore Wind und Offshore Wind beträgt auf 329.300 GWh. Davon etwa 24.500 GWh entsprechen zu der endlichen Ausfallenergie (Curtailment) des Systems, während eine Strommenge von 8.100 GWh gespeichert wird.

04.03:

- I realised a discreapency in 2050 input data

Site	Process	inv-cost	fix-cost	var-cost	wacc	depreciation	area-per-cap
Germany	Biomass	740.000	22.200	1	0	25	#NV
Germany	CCGT New	800.000	30.000	4	0	40	#NV
Germany	Biogas	1.110.000	44.400	1	0,027	33	#NV

		Biogas	
	Einheit	2015	2035
Investitionskosten	$€_{2012}$/kW	3.500	3.100
Stromerzeugung	MWh/MW	7.000	6.000
Fixe Betriebskosten	% der Invest.-kosten	4%	4%
Variable Betriebskosten	$€_{2012}$/MWh	1	1
Wirkungsgrad	%	33%	37%
CO_2-Emissionen	g/kWh	0	0
WACC (real)	%	6,5%	6,5%

		Biomasse	
	Einheit	2015	2035
Investitionskosten	$€_{2012}$/kW	2.500	2.200
Stromerzeugung	MWh/MW	6.500	5.500
Fixe Betriebskosten	% der Invest.-kosten	3%	3%
Variable Betriebskosten	$€_{2012}$/MWh	1	1
Wirkungsgrad	%	33%	37%
CO_2-Emissionen	g/kWh	0	0
WACC (real)	%	6,5%	6,5%

Erdgas und Biomasse

	GuD-Kraftwerk	Biomasse-kraftwerk	Biogas-Kraftwerk
Brennstoffkosten in €/MWh_{th}	34 (4)	28 (8)	28 (8)
Investitionskosten in €/MW_{el}	800.000 (3)	2.000.000 (8)	3.000.000 (8)
Fixkosten in €/MW_{el}	30.000 (3)	60.000 (8)	44,400 (8)
Variable Kosten in €/MWh_{el}	4,0 (4)	1,0 (8)	1,0 (8)
Kapitalzinssatz %	0,052 (4)	0.027 (8)	0.027 (8)
Abschreibungsdauer in Jahren	33 (1)	25 (1)	33 (1)
Installierte Leistung in MW (Norden)	5.654 (9)	-	-
Installierte Leistung in MW (Süden)	2.346 (9)	-	-
Verfürgbare Primärenergie in MWh (Norden)	-	128.000.000 (10)	38.726.000 (6)
Verfürgbare Primärenergie in MWh (Süden)	-	134.000.000 (10)	33.507.000 (6)
Effizienz (%)	64% (1)	37% (8)	37% (8)
CO2-Austoß in t/GWh_{th}	201,6 (1)	0	0

Rahmenannahmen für die Kosten konventioneller Kraftwerke, 2030 und 2050

Tabelle 3-2

	Investitionskosten [€/kW]		Fixe Betriebs-kosten [€/kW]	Lebensdauer [a]	Anmerkung
	2030	2050			
Braunkohlekraftwerk	1.600	1.600	40	40	
Steinkohlekraftwerk	1.300	1.300	40	40	
GuD-Kraftwerk	800	800	30	40	
Gasturbine	400	400	10	20	
Wasserkraftwerk	1.000	1.000	40	45	erschlossener Standort

Berechnungen und Schätzungen des Öko-Instituts

Down all values are noted with SI decimals

Biogas----------Thesis----multiplication-------------------EXCEL-------------------------------------RESOURCE

InvCosts: 3,000,000---- x 0.37--------------------------------- 1,110,000 3,100,000 Euros/MW

FixCosts: 44.400--------3,000,000 x0.37 x 0.04-------------44,000 124,000 Euros/MW (invCost x 4%) is that annual?

VarCosts: 1 ---1 1

Efficiency: 0.37

Total hours of production annually : 6000h -------------> 68,5% capacity factor

Biomass---------Thesis----multiplication-------------------EXCEL-------------------------------------RESOURCE

InvCosts: 2,000,000---- x 0.37--------------------------------------- 740,000 2,200,000 Euros/MW

FixCosts: 60,000------- x0.37 ---------------------------------------44,000 66,000 Euros/MW (invCost x 3%) is that annual?

VarCosts: 1 --1 1

Efficiency: 0.37

Total hours of production annually : 5500h -------------> 63% capacity factor

GuD---------Thesis----multiplication-------------------EXCEL-------------------------------------RESOURCE

InvCosts: 800,000-- 800,000 800,000 Euros/MW

FixCosts: 30,000---30,000 30,000 Euros/MW

VarCosts: 4 ---4 ?

Efficiency: 0.64 ?

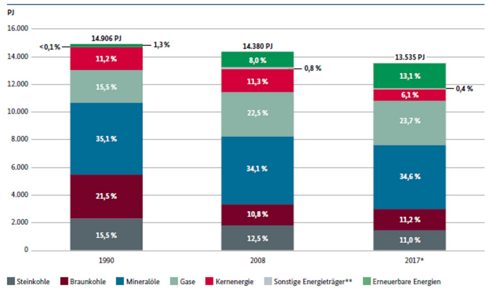
Total hours of production annually : 3500h -------------> 40% capacity factor

Die Stromgestehungskosten von GuD-Kraftwerken haben eine Bandbreite zwischen 7,78 und 9,96 €Cent/kWh

Aus <https://www.ise.fraunhofer.de/content/dam/ise/de/documents/publications/studies/DE2018_ISE_Studie_Stromgestehungskosten_Erneuerbare_Energien.pdf>

-Onun disinda BMWi reportundan:

Abbildung 4: Primärenergiemix in Deutschland 1990, 2008 und 2017



* vorläufige Angaben
** Sonstige Energieträger: Nichterneuerbare Abfälle, Abwärme, Strom- und Fernwärmeaustauschsaldo

Quelle: Eigene Darstellung UBA auf Basis AGEB, Auswertungstabellen, Stand 09/2017; AGEB, Energieverbrauch 2017, Stand 02/2018

11.03.2020
- Conferance journal bakmaca

Renewable and sustainable energy reviews
Energy policy
Applied energy
Nature energy

IEW conferance 2020 Freiburg Closed for 2020 Closed already http://www.internationalenergyworkshop.org/meetings-13.html

EMPE 2020 european commission :http://www.energymodellingplatform.eu/ Not opened yet

- Cocugun input fileini kontrol edip input file hazirlamaca

Bulunmasi gereken infolar:
- ~~2020 hourly demand (en kötü 2017 demandi alinabilir)~~

2019 hourly demand is taken from smard
Load is taken from: https://www.smard.de/home/downloadcenter/download_marktdaten/726#!?downloadAttributes=%7B%22selectedCategory%22:2,%22selectedSubCategory%22:5,%22selectedRegion%22:%22DE%22,%22from%22:1546297200000,%22to%22:1577918759999,%22selectedFileType%22:%22XLS%22%7D

15 minute resolution realtime demand from 01.01.2019 to 01.01.2020 processed with python and resolution decreased to 1 hour.

- Installed capacities for coal and nuclear: https://www.smard.de/home/marktdaten/78?marketDataAttributes=%7B%22resolution%22:%22month%22,%22from%22:1514761200000,%22to%22:1577919599999,%22moduleIds%22:%5B3000189,3003792,3004076,3000186,3000188,3000194,3004073,3004072,3004075,3000198,3004074,3000207%5D,%22selectedCategory%22:3,%22activeChart%22:false,%22style%22:%22color%22,%22region%22:%22DE%22%7D#chart-legend

Installierte Netto-Engpassleistung von Anlagen mit mind. 1 MW installierter Leistung [MW]. Es werden ausschließlich bereits in Betrieb genommene Anlagen berücksichtigt. Die Daten beziehen sich auf den 1. Januar des jeweiligen Jahres. Angaben zur installierten Netto-Engpassleistung erneuerbarer Energieträger werden im August des jeweiligen Jahres aktualisiert. [Quelle: ENTSO-E]
Link zum Artikel

Stromerzeugung - Installierte Erzeugungsleistung

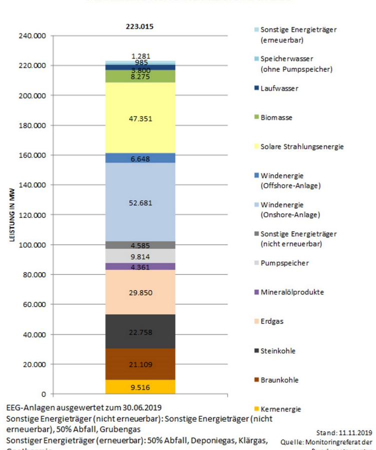
Weitere Kategorien anzeigen

Startzeitpunkt	Biomasse in MW	Wasserkraft in MW	Wind Offshore in MW	Wind Onshore in MW	Photovoltaik in MW	Sonstige Erneuerbare in MW	Kernenergie in MW	Braunkohle in MW	Steinkohle in MW	Erdgas in MW	Pumpspeicher in MW	Sonstige Konventionelle in MW
01.01.2018, 00:00	7.396	5.300	5.051	51.633	42.804	534	9.516	21.275	25.035	31.361	8.918	7.375
01.01.2019, 00:00	7.752	5.281	6.393	52.792	45.299	487	9.516	21.205	25.293	31.664	9.422	7.277
01.01.2020, 00:00	7.855	5.256	7.709	53.405	46.471	523	8.114	21.067	22.458	31.712	9.422	7.592

Aus <https://www.smard.de/home/marktdaten/78?marketDataAttributes=%7B%22resolution%22:%22month%22,%22from%22:1514761200000,%22to%22:1577919599999,%22moduleIds%22:%5B3000189,3003792,3004076,3000186,3000188,3000194,3004073,3004072,3004075,3000198,3004074,3000207%5D,%22selectedCategory%22:3,%22activeChart%22:false,%22style%22:%22color%22,%22region%22:%22DE%22%7D>

Kraftwerkliste excel

INSTALLIERTE NETTO-NENNLEISTUNG IN DEU

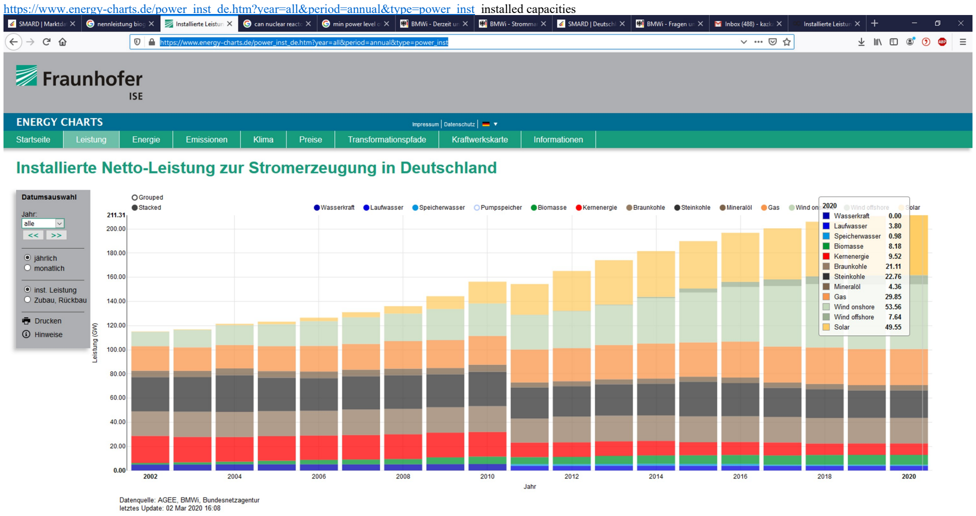


……What has changed…. All installed capacities are increased . Erdgas capacitty written to dummy gas and it has a drastic increased compared to old values. Sonstige erneubasre and sonstige conventionale are skipped. Also

Biogass 5228MW https://de.statista.com/statistik/daten/studie/167673/umfrage/installierte-elektrische-leistung-von-biogasanlagen-seit-1999/

5620 MWel https://www.foederal-erneuerbar.de/uebersicht/bundeslaender/BW%7CBY%7CB%7CBB%7CHB%7CHH%7CHE%7CMV%7CNI%7CNRW%7CRLP%7CSL%7CSN%7CST%7CSH%7CTH%7CD/kategorie/bioenergie/auswahl/270-installierte_leistun/

https://www.energy-charts.de/power_inst_de.htm?year=all&period=annual&type=power_inst installed capacities



- Life time for coal and nuclear: 2038
- Max grad for coal and lignite plant (?) because in original input file max grad is defined for those
-
- Min load fraction for nuclear
- Wacc take real values not nominal values
- Depreciation time for new technologies
- CO2 budget value
- Discount rate
- Commodity Prices
- In out ratio for coal nuclear and lignite
- STORAGE, DSM, Timevarefficiency ??????
- SupIm will be taken same as 2017 values but check first
- Electricity buy sell price not modeled (Cant be taken from original file)
-

Agora energiewende diye bi organizasyon sirket buldum. Oradan timeserieslere ulasabilecegimi düsünüyorum.

- Kodu intertemporale göre ayarlamaca

16.03

- Depreciation of current pps
  (Kraftwerksstilllegungsanzeigen)
  Kraftwerksstilllegungsanzeigenliste (KWSAL)
  Aus <https://www.govdata.de/suchen/-/details/kraftwerksstilllegungsanzeigenliste-kwsal>
  Aus <https://www.bundesnetzagentur.de/DE/Sachgebiete/ElektrizitaetundGas/Unternehmen_Institutionen/Versorgungssicherheit/Erzeugungskapazitaeten/KWSAL/KWSAL_node.html;jsessionid=2BA84F918AFFC67DC31DCFBC5E94B593>
- German powerplant list from 2012 to 2019 there is also another excell which shows planned depreciations it is in input resources folder
  https://www.bundesnetzagentur.de/DE/Sachgebiete/ElektrizitaetundGas/Unternehmen_Institutionen/Versorgungssicherheit/Erzeugungskapazitaeten/Kraftwerksliste/kraftwerksliste-node.html;jsessionid=2BA84F918AFFC67DC31DCFBC5E94B593

- Bu listeden aldigimiz degerler nennleistung :
  Die Nennleistung ist die (meist) vom Hersteller genannte Orientierungshilfe, mit der eine Anlage dauerhaft wirtschaftlich hinsichtlich Verschleiß und Energieeinsatz sicher betrieben werden kann. Die installierte Leistung beschreibt hingegen die Spitzenleistung einer Maschine, beispielsweise eines Generators oder einer photoelektrischen Zelle. Mit installierter Leistung können in der Energiewirtschaft aber auch die zusammengefassten Spitzenleistungen mehrerer Stromerzeugungseinheiten gemeint sein.

  Daneben existiert eine Bemessungsleistung für Anlagen nach dem EEG, die allerdings vor allem bilanziell zu verstehen ist. Sie ergibt sich aus den effektiv produzierten Kilowattstunden geteilt durch Summe aller Jahresstunden und dient zur Berechnung der Vergütung nach dem Erneuerbaren-Energien-Gesetz (EEG).

18.03
- Bugün Co2 budget ile ugrasacagim (2 saat max) sonra bulöduklarimizi degerlendiririz ve özetleriz.

23.03
-Bugün modeli intertemporal yapmak ile ugrasacagim
-cost reporting does not consider different years is that normal.
-Is capacity reporting works fine
-Objective yer yil icin ayri site belirtiyo onu düzelt

- Ok sikinti unpack objective diette, o sitelari sürekli basiyor ama bunu düzeltmeden önce bunu degistirmek baska problem yaratirmi bakman lazim
'Photovoltaics': ['Mid', 'North', 'South', 'Mid', 'North', 'South', 'Mid', 'North', 'South', 'Mid', 'North', 'South'], 'Wind park': ['Mid', 'North', 'South', 'Mid', 'North', 'South', 'Mid', 'North', 'South', 'Mid', 'North', 'South']}

```
objective_sites=list(set(data['site'].index.get_level_values(1)))
objective_dict[objective_pro]=objective_sites
```

- Stack plotlarda x label biraz daha asagida durmali tixler ile üstüste biniyor
- Capacity data frame is tidied for easier plotting and statistics
- Change column names with slack percentages instead of decimal percentages
- -model intertemporal icin calisiyor yarin line plot da ayarlandi
- Yarin ise intertemporal icin costlari dogru arttiriyor mu bakarak basla ayrica kapasiteler de düzgün minimize maximize edilmismi ona bak
- Daha sonra input filea dön

- Co2 budgeti özetle

24.03

~~Intertemporalde costlar bütün slackler icin ayni cikiyor. Bunun böyle olmamasi lazim~~

~~Check:~~

- ~~If it is working well with single year: no it is not working~~
- ~~Reporting: + reporting is okay the data frame has wrong values~~
- ~~Forming data frame~~
- ~~Constraint~~

~~For single year:~~
~~Costs after cost minimization~~

~~prob.costs.get_values()~~
~~Out[8]:~~
~~{'Invest': 9000966255.321009,~~
~~'Fixed': 2356620096.347297,~~
~~'Variable': 784283990.0256143,~~
~~'Fuel': 11198199785.210905,~~
~~'Environmental': 0.0,~~
~~'Revenue': 0.0,~~
~~'Purchase': 4845594.0}~~

~~prob.opt_cost_sum~~
~~Out[22]: 23344915720.904823~~
~~23344915720,904823~~

~~prob.cap_pro.display()~~
~~cap_pro : Size=25~~

```
Key                                  : Value
(2020, 'Mid', 'Biomass plant') :          5000.0
(2020, 'Mid', 'Gas plant') : 44929.984670904254
(2020, 'Mid', 'Hydro plant') :          1400.0
(2020, 'Mid', 'Lignite plant') : 22121.902641716733
(2020, 'Mid', 'Photovoltaics') :           0.0
(2020, 'Mid', 'Slack powerplant') :       999999.0
(2020, 'Mid', 'Wind park') :          13000.0
(2020, 'North', 'Biomass plant') :          0.0
(2020, 'North', 'Coal plant') :           0.0
(2020, 'North', 'Feed-in') :          2500.0
(2020, 'North', 'Gas plant') :           0.0
(2020, 'North', 'Hydro plant') :         20000.0
(2020, 'North', 'Photovoltaics') :          0.0
(2020, 'North', 'Purchase') :          2500.0
(2020, 'North', 'Slack powerplant') :       999999.0
(2020, 'North', 'Wind park') :           0.0
(2020, 'South', 'Biomass plant') :          0.0
(2020, 'South', 'Coal plant') :           0.0
(2020, 'South', 'Feed-in') :          1500.0
(2020, 'South', 'Gas plant') :           0.0
(2020, 'South', 'Hydro plant') :           0.0
(2020, 'South', 'Photovoltaics') : 21111.192283473432
(2020, 'South', 'Purchase') :          1500.0
(2020, 'South', 'Slack powerplant') :       999999.0
(2020, 'South', 'Wind park') :           0.0
```

```
cost_type
Invest          9.000966e+09
Fixed           2.356620e+09
Variable        7.842840e+08
Fuel            1.119820e+10
Environmental   0.000000e+00
Revenue         0.000000e+00
Purchase        4.845594e+06
Name: costs, dtype: float64
sum(costs)
Out[6]: 23344915720.904823
```

```
                                Total          New
Stf  Site  Process
2020 Mid   Biomass plant      5000.000000   5000.000000
           Gas plant         44929.984671  44929.984671
           Hydro plant        1400.000000   1400.000000
           Lignite plant     22121.902642  22121.902642
           Photovoltaics         0.000000      0.000000
           Slack powerplant 999999.000000      0.000000
           Wind park         13000.000000  13000.000000
     North Biomass plant         0.000000      0.000000
           Coal plant            0.000000      0.000000
           Feed-in            2500.000000   2500.000000
           Gas plant             0.000000      0.000000
           Hydro plant       20000.000000  20000.000000
           Photovoltaics         0.000000      0.000000
           Purchase           2500.000000   2500.000000
           Slack powerplant 999999.000000      0.000000
           Wind park             0.000000      0.000000
     South Biomass plant         0.000000      0.000000
           Coal plant            0.000000      0.000000
           Feed-in            1500.000000   1500.000000
           Gas plant             0.000000      0.000000
           Hydro plant           0.000000      0.000000
           Photovoltaics     21111.192283  21111.192283
           Purchase           1500.000000   1500.000000
           Slack powerplant 999999.000000      0.000000
           Wind park             0.000000      0.000000
```

~~.Cost restrict rule is defined:~~

```
res_cost_restrict_rule(m)
assertm.cost_factor>=0,"slackvalueisnotdefinedproperly.Slackvaluemustbeapositivenumber."
return(1+m.cost_factor)*m.opt_cost_sum==pyomo.summation(m.costs)
```

~~Slack loop:~~

~~First Slack: 0,02~~ . . . .

~~Minimization:~~

~~prob.costs.get_values()~~
~~Out[18]:~~
~~{'Invest': 8849566717.73564,~~
~~'Fixed': 2298584159.9855847,~~
~~'Variable': 818184617.8202662,~~
~~'Fuel': 11840632945.781431,~~
~~'Environmental': 0.0,~~
~~'Revenue': 0.0,~~
~~'Purchase': 4845594.0}~~

~~sum(prob.costs.get_values().values())~~
~~Out[21]: 23811814035.32292…. This value is correct for costs~~

```
cap_pro : Size=25
    Key                                   : Value
        (2020, 'Mid', 'Biomass plant') :             5000.0
            (2020, 'Mid', 'Gas plant') :  47985.69795367881
          (2020, 'Mid', 'Hydro plant') :             1400.0
        (2020, 'Mid', 'Lignite plant') : 20594.046000329457
        (2020, 'Mid', 'Photovoltaics') :                0.0
     (2020, 'Mid', 'Slack powerplant') :           999999.0
            (2020, 'Mid', 'Wind park') : 11508.703508279323
      (2020, 'North', 'Biomass plant') :                0.0
         (2020, 'North', 'Coal plant') :                0.0
            (2020, 'North', 'Feed-in') :             2500.0
          (2020, 'North', 'Gas plant') :                0.0
        (2020, 'North', 'Hydro plant') :            20000.0
      (2020, 'North', 'Photovoltaics') :                0.0
           (2020, 'North', 'Purchase') :             2500.0
   (2020, 'North', 'Slack powerplant') :           999999.0
          (2020, 'North', 'Wind park') :                0.0
      (2020, 'South', 'Biomass plant') :                0.0
         (2020, 'South', 'Coal plant') :                0.0
            (2020, 'South', 'Feed-in') :             1500.0
          (2020, 'South', 'Gas plant') :                0.0
        (2020, 'South', 'Hydro plant') :                0.0
      (2020, 'South', 'Photovoltaics') :  21111.19248672262
           (2020, 'South', 'Purchase') :             1500.0
   (2020, 'South', 'Slack powerplant') :           999999.0
          (2020, 'South', 'Wind park') :                0.0
Out[23]: NoneType
```

~~In excel cap_pro values seems correct but costs are not read correctly~~

costs, cpro, ctra, csto = get_constants(prob)

```
cost_type
Invest          9.000966e+09
Fixed           2.356620e+09
Variable        7.842840e+08
Fuel            1.119820e+10
Environmental   0.000000e+00
Revenue         0.000000e+00
Purchase        4.845594e+06
Name: costs, dtype: float64
```

```
                                 Total           New
Stf  Site  Process
2020 Mid   Biomass plant       5000.000000   5000.000000
           Gas plant          47985.697954  44929.984671
           Hydro plant         1400.000000   1400.000000
           Lignite plant      20594.046000  22121.902642
           Photovoltaics          0.000000      0.000000
           Slack powerplant  999999.000000      0.000000
           Wind park          11508.703508  13000.000000
     North Biomass plant          0.000000      0.000000
           Coal plant             0.000000      0.000000
           Feed-in             2500.000000   2500.000000
           Gas plant              0.000000      0.000000
           Hydro plant        20000.000000  20000.000000
           Photovoltaics          0.000000      0.000000
           Purchase            2500.000000   2500.000000
           Slack powerplant  999999.000000      0.000000
           Wind park              0.000000      0.000000
     South Biomass plant          0.000000      0.000000
           Coal plant             0.000000      0.000000
           Feed-in             1500.000000   1500.000000
           Gas plant              0.000000      0.000000
           Hydro plant            0.000000      0.000000
           Photovoltaics      21111.192487  21111.192283
           Purchase            1500.000000   1500.000000
           Slack powerplant  999999.000000      0.000000
           Wind park              0.000000      0.000000
```

ok get constants cpro works but costs do not work fucking bitch

~~Maximization:~~

~~prob.costs.get_values()~~
~~Out[29]:~~

Get_constants

~~Maximization:~~

~~prob.costs.get_values()~~
~~Out[29]:~~
~~{'Invest': 12028900643.434425,~~
~~'Fixed': 2993178644.0300083,~~
~~'Variable': 623938609.0510759,~~
~~'Fuel': 8160950544.807412,~~
~~'Environmental': 0.0,~~
~~'Revenue': 0.0,~~
~~'Purchase': 4845594.0}~~

~~sum(prob.costs.get_values().values())~~
~~Out[30]: 23811814035.32292~~

Get_constants

```
cost_type
Invest           9.000966e+09
Fixed            2.356620e+09
Variable         7.842840e+08
Fuel             1.119820e+10
Environmental    0.000000e+00
Revenue          0.000000e+00
Purchase         4.845594e+06
Name: costs, dtype: float64
```

```
cap_pro : Size=25
    Key                                   : Value
         (2020, 'Mid', 'Biomass plant') :             5000.0
             (2020, 'Mid', 'Gas plant') : 30483.403428497997
           (2020, 'Mid', 'Hydro plant') :             1400.0
         (2020, 'Mid', 'Lignite plant') : 29345.193262919896
         (2020, 'Mid', 'Photovoltaics') :                0.0
      (2020, 'Mid', 'Slack powerplant') :           999999.0
             (2020, 'Mid', 'Wind park') :            13000.0
       (2020, 'North', 'Biomass plant') :                0.0
          (2020, 'North', 'Coal plant') :                0.0
             (2020, 'North', 'Feed-in') :             2500.0
           (2020, 'North', 'Gas plant') :                0.0
         (2020, 'North', 'Hydro plant') :            20000.0
       (2020, 'North', 'Photovoltaics') :                0.0
            (2020, 'North', 'Purchase') :             2500.0
    (2020, 'North', 'Slack powerplant') :           999999.0
           (2020, 'North', 'Wind park') : 12712.994506191966
       (2020, 'South', 'Biomass plant') :                0.0
          (2020, 'South', 'Coal plant') :                0.0
             (2020, 'South', 'Feed-in') :             1500.0
           (2020, 'South', 'Gas plant') :                0.0
         (2020, 'South', 'Hydro plant') :                0.0
       (2020, 'South', 'Photovoltaics') : 21111.192283473436
            (2020, 'South', 'Purchase') :             1500.0
    (2020, 'South', 'Slack powerplant') :           999999.0
           (2020, 'South', 'Wind park') :                0.0
```

```
                                       Total            New
Stf  Site  Process
2020 Mid   Biomass plant       5000.000000    5000.000000
           Gas plant          30483.403428   44929.984671
           Hydro plant         1400.000000    1400.000000
           Lignite plant      29345.193263   22121.902642
           Photovoltaics          0.000000       0.000000
           Slack powerplant  999999.000000       0.000000
           Wind park          13000.000000   13000.000000
     North Biomass plant          0.000000       0.000000
           Coal plant             0.000000       0.000000
           Feed-in             2500.000000    2500.000000
           Gas plant              0.000000       0.000000
           Hydro plant        20000.000000   20000.000000
           Photovoltaics          0.000000       0.000000
           Purchase            2500.000000    2500.000000
           Slack powerplant  999999.000000       0.000000
           Wind park          12712.994506       0.000000
     South Biomass plant          0.000000       0.000000
           Coal plant             0.000000       0.000000
           Feed-in             1500.000000    1500.000000
           Gas plant              0.000000       0.000000
           Hydro plant            0.000000       0.000000
           Photovoltaics      21111.192283   21111.192283
           Purchase            1500.000000    1500.000000
           Slack powerplant  999999.000000       0.000000
           Wind park              0.000000       0.000000
```

~~Ok as understood model works fine just get constants function can not call costs properly. Let me look at this~~

~~-first look if get_consrant is being utilized somewhere important.~~

```
▼ Function
    get_entity(instance, name)
▼ Found usages 22 usages
  ▼ Unclassified usage 17 usages
    ▼ urbs 17 usages
      ▼ nopt 15 usages
        ▶ output_nopt.py 10 usages
        ▶ plot_nopt.py 3 usages
        ▶ pyomoio_nopt.py 1 usage
        ▶ saveload_nopt.py 1 usage
      ▶ nopt\features_nopt 2 usages
  ▶ Usage in import statement 5 usages
```

~~Already being used a lot: cost is called only by my function and report function but I do not know if it works fine for all types of entities.~~

Cost:

```
▼ get_constants 1 usage
    34 costs = get_entity(instance, 'costs')
▼ get_timeseries 9 usages
    85 timesteps = sorted(get_entity(instance, 'tm').index)
    113 eco = get_entity(instance, 'e_co_stock')
    122 created = get_entity(instance, 'e_pro_out')
    131 consumed = get_entity(instance, 'e_pro_in')
    148 imported = get_entity(instance, 'e_tra_out')
    171 exported = get_entity(instance, 'e_tra_in')
    221 dsmup = get_entity(instance, 'dsm_up')
    222 dsmdo = get_entity(instance, 'dsm_down')
    266 voltage_angle = get_entity(instance, 'voltage_angle')
```

```
▼ plot_nopt.py 3 usages
  ▼ plot 1 usage
      501 timesteps = sorted(get_entity(prob, 'tm').index)
  ▼ result_figures 2 usages
      756 dt = get_entity(prob, 'dt')
      764 periods = {'all': sorted(get_entity(prob, 'tm').index)}
▼ pyomoio_nopt.py 1 usage
  ▼ get_entities 1 usage
      136 other = get_entity(instance, name)
▼ saveload_nopt.py 1 usage
  ▼ create_result_cache 1 usage
      16 result_cache[entity] = get_entity(prob, entity)
```

```
▼ nopt\features_nopt 2 usages
  ▼ modelhelper.py 2 usages
    ▼ read_capacity 1 usage
        224 cpro = get_entity(m, 'cap_pro')
    ▼ read_costs 1 usage
        238 costs = get_entity(m, 'costs')
```

~~Ok this function has a short circuit .~~

```
ifhasattr(instance,'_result')andnameininstance._result:
returninstance._result[name].copy(deep=True)
```

~~Thisis disturbing us, result is created when h5 file is saved but I save it just when cost optimum is solved. I am going to create this result each time I solve the problem.~~
~~I can also delete the short circuit~~

~~-result is created in saveload.py~~

```
defsave(prob,filename):
"""Saveurbsmodelinputandresultcachetoa HDF5storefile.

Args:
-prob:aurbsmodelinstancecontainingasolution
-filename:HDF5storefiletobewritten

Returns:
Nothing
"""
importwarnings
importtables
warnings.filterwarnings('ignore',
category=pd.io.pytables.PerformanceWarning)
warnings.filterwarnings('ignore',
category=tables.NaturalNameWarning)
```

```
ifnothasattr(prob,'_result'):
prob._result=create_result_cache(prob)
```

```
withpd.HDFStore(filename,mode='w')asstore:
fornameinprob._data.keys():
store['data/'+name]=prob._data[name]
fornameinprob._result.keys():
store['result/'+name]=prob._result[name]
```

~~Ok simdi amac bu _result variableini yeniden olusturmak:~~

~~prob._result['costs']~~
~~Out[10]:~~
~~cost_type~~
~~Invest 9.000966e+09~~
~~Fixed 2.356620e+09~~
~~Variable 7.842840e+08~~
~~Fuel 1.119820e+10~~
~~Environmental 0.000000e+00~~
~~Revenue 0.000000e+00~~
~~Purchase 4.845594e+06~~
~~Name: costs, dtype: float64~~

~~cost_type~~
~~Invest 8.849567e+09~~
~~Fixed 2.298584e+09~~
~~Variable 8.181846e+08~~
~~Fuel 1.184063e+10~~
~~Environmental 0.000000e+00~~
~~Revenue 0.000000e+00~~
~~Purchase 4.845594e+06~~
~~Name: costs, dtype: float64~~

YASSSSSS!!!!!!!

-So what I have done:

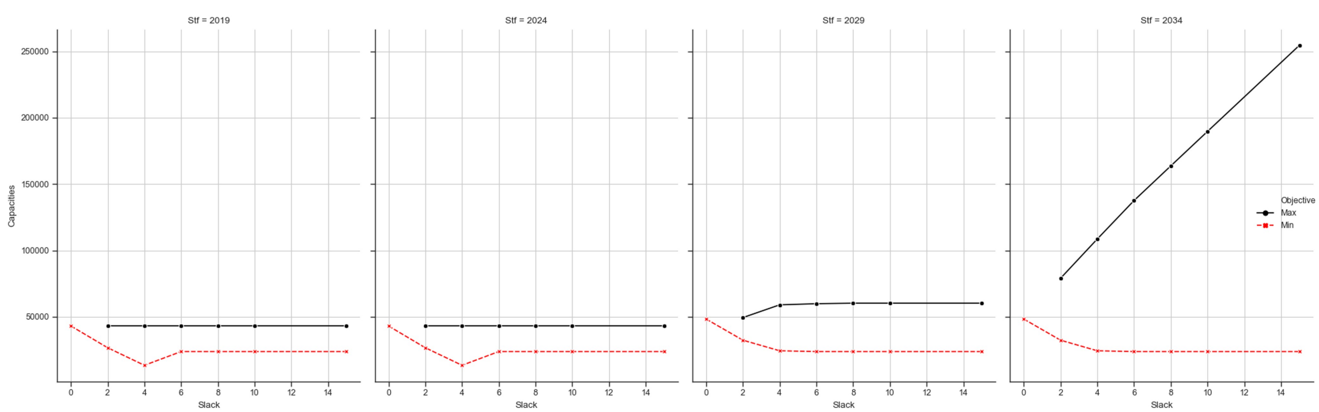
I changed the short circuit argument in get_entities because get_result_cache uses get_entities and get entities uses results cache if exixtes.

```
#magic:short-circuitifproblemcontainsaresultcache
ifhasattr(instance,'_result')andnameininstance._resultand('cost'ininstance.objective_dict.keys()or'CO2'ininstance.objective_dict.keys()):
returninstance._result[name].copy(deep=True)
```

Then I added
```
prob._result=create_result_cache(prob)
```

After each solution so that my _result cache will also be up to date in case other functionalities uses this cache to read values

-But there is a problem with results: only the last year behaves as we wanted since we minimized only the last year. Capacity of year 2019 slack 6 is equal to 2034 slack 6



-Now CO2 values summary +

30.03.2020-31.03.2020

- Start with midterm presentation
- Yarin bulduklarini excelle yerlestir. Sunum icin güzel ciktilar al görselleri hazirla soner ile tartis

02.04
- Today I will do sth completely different and try to learn plotting deeply

13.04

- Today I go back to where I was
- My plan is to do a time line at the end of the day
- Also I start forming 2020 file
- First I will start with reading the file Soner sent me last week
- And I will write down all the values that needs to be proven +
- Check the problem with geothermal in 2050b data

'Dynamische und intersektorale Maßnahmenbewertung zur kosteneffizienten Dekarbonisierung des Energiesystems'

- I can add Carbon storage as a process and try to neutralize co2 emmission till 2050

  CCS:

Maximum technical reduction potential: 362 million t CO_2 / a Assumptions: German storage potential of 8.75 - 15.75 billion t CO_2 (2.75 billion t. CO_2 (natural gas deposits) and 6 - 13 billion t. CO_2 (salinare Aquifere)) / BGR-02 10 / Assumed 33 years from 2017 to 2050 result in a technical potential of 265 - 477 million t CO_2 / a Maximum balance sheet coverage 2014: 132% Assumptions: According to the above-mentioned maximum saving potential of 477 million t CO_2 / a results in an additional annual reduction potential of 115 million t CO_2

Hydro- ror:

Maximum technical reduction potential: 45 million t CO_2 / a Assumptions: Potential for Germany according to / MKA-01 14 /: 92.6 TWh / a Partial substitution of the German electricity mix 2014 Emission factor electricity mix 2014: 537 g / kWh Maximum balance cover 2014: 15 % Assumptions: based on electricity generation in 2014

PV(dachflasche):

Maximum technical reduction potential: 88 million t CO_2 / a

Roof area potential: 194 GW

Average full load hours PV in Germany: 945 h / a in our file 1093 that is a 149 hour difference which is equal to 6 days and 5 hours

Resulting electricity generation: 183 TWh (FfE-internal calculations with regional data from the FREM database)

Partial substitution of the German electricity mix 2014 Emission factor electricity mix 2014: 537 g / kWh Maximum balance cover 2014: 29% Assumptions: based on the electricity generation 2014

PV(farms):

Maximales technisches Verminderungspotenzial: 302 Mio. t CO_2/a
Freiflächenpotenzial: 935 GW
Durchschnittliche Volllaststunden Freiflächen PV in Deutschland: 1.052 h
Resultierende Stromerzeugung: 984 TWh (FfE-interne Berechnungen mit Regionaldaten der Datenbank FREM)
Vollständige Substitution des deutschen Strommix 2014 Emissionsfaktor Strommix 2014: 537 g/kWh
Maximale bilanzielle Deckung: 157 %
Annahmen: bezogen auf die Stromerzeugung 2014

Wind (onshore)

Maximales technisches Verminderungspotenzial: 94 Mio. t CO_2/a
Abschätzung des maximalen Potenzials auf Basis interner Berechnungen 130 GW /FFE-21 17/
Durchschnittliche Volllaststunden Wind-Onshore in Deutschland: 1.500 h /AEE-02 13//AEE-02 13/ for our paper 1559
Resultierende Stromerzeugung 195 TWh /FFE-21 17/

Emissionsfaktor Strommix 2014: 537 g/kWh.
Der gesamte Strombedarf Deutschlands könnte gedeckt werden Maximale bilanzielle Deckung 2014: 31 %

Biomass-to-Liquid (BtL):

As the technical potential of biomass in Germany is only sufficient for a partial substitution of fossil fuels,
it is assumed that the substitution is mainly used in the transport sector, which is difficult to decarbonize,
and where fossil liquid fuels (i.e. diesel, petrol and kerosene) are substituted.
These are substituted by the synthetic BtL fuel according to their share in the total emissions.
The reduction potential results from this logic.
Maximum balance sheet coverage in 2014: 21%
assumptions: based on the total demand of 639 TWh for diesel, gasoline and kerosene in 2014 according to / FFE-13 17 /

Addition of biomass in conv. Power plants:

The sustainable wood potential in Germany is 190 TWh / a / AEE-03 14 /, so that there is a coverage of 30%.

GuD and Erdgas:

Maximum technical reduction potential: 69 million t CO_2 / a
Assumptions:
Electrical efficiency: 43%
Thermal efficiency: 49%
Emission factor natural gas: 0.0559 t CO_2 / PJ
Fuel requirement: 5,256 PJ
It is assumed that the technology will replace all conventional electricity generation
and the heat generated in this way is fed into existing district heating networks and completely displaces the technologies established there.
The savings are achieved through the better efficiency of the substitution technology and the use of the less emission-intensive energy source natural gas.

Tabelle 1-4: PV Freifläche

	Einheit	2020	2030	2040	2050	Quelle
Nutzungsdauer	a	25	25	25	25	/DENA-01 18/ Seite 428
Wirkungsgrad elektrisch	%	17 %	21 %	26 %	30 %	/ISE-01 15/ Seite 20
CAPEX	€/ MW_{el_out}	827.057	660.122	527.204	445.000	Mittelwert aus /AGORA-03 17/ Seite 19, /AGORA-02 15/ Seite 50, /DIW-07 13/ Seite 9, /RLI-01 13/ Seite 69, /DENA-01 18/ Seite 428 und /AGORA-03 17/ Seite 19
fixe OPEX	€/ (MW_{el_out}*a)	16.541	13.202	10.544	8.900	/AGORA-03 17/ Seite 19

CapEx refers to a Capital expenditure while
OpEx refers to an Operational expenditure.

Capital expenditure is incurred when a business acquires assets that could be beneficial beyond the current tax year. For instance, it might buy brand new equipment or buildings. Also, it could upgrade an existing asset to boost its value beyond the current tax year. CapEx is also known as a Capital expense.

They are the costs that a business incurs while in the process of turning its inventory into an end product. Hence, depreciation of fixed assets that are used in the production process is considered OpEx expenditure. OpEx is also known as an operating expenditure, revenue expenditure or an operating expense.

CapEx – Capital expenditures are not fully deducted in the accounting period they were incurred. In other words, they are not fully subtracted from the revenue when computing the profits or losses a business has made. However, intangible assets are amortized over their lifespan while the tangible ones are depreciated over their life cycle. All monies spent to get new inventory, including machinery or intellectual property, are grouped under CapEx spendings.

- Purchase of fixed assets.
- Preparation of the purchased asset so it can be appropriate for business use.
- Fixing of asset's problems,
- Restoration of an asset's value through upgrading
- Adapting a machine to a different use

OpEx – Operating expenses are fully deducted in the accounting period they were incurred. All funds spent when converting inventory into throughput falls under OpEx. This includes employee wages, repair and maintenance of equipment, rental fees, and utility bills and so on. If a business invests in real estate, these spending is approved as CapEx budget and the expense is grouped under CapEx. However, all the costs incurred when managing such an income generating building falls under OpEx.
- License fees
- Advertising costs
- Legal and attorney fees

- Telephone and power overheads
- Insurance fees
- Property management costs
- Property taxation expenses
- Vehicle fuel and repair costs
- Leasing commissions
- Salary and wages
- Raw materials and supplies
- Office overheads

14.04

-Guideline for dependent investment costs and capacities:

Investment costs:

Biomass:

Biomass_in	1
Elec_out	0.37

For biomass
1MW_el_out -----> 740,000 Euro
1MW_bm_in ----> 740,000 x 0.37

1MW_bm_in =1MW process capacity

For 1 MW process capacity 740,000 Euro is needed

Capacities:

Well:

G_in	1
WW_out	0.7

Geop:

WW_in	1
Elec_out	0.14

1MW_G_in =0.7 MW_ww_out
----> 1MW_ww_in =1MW geop capacity =1/0.7 MW MW_G_in =1/0.7 MW well capacity

------> (1/0.7)MW well =1 Mwgeo

Dagha fazla well capacitysi olacak yani

1MW Geop capacity = 1MW_ww_in

1MW Well capacity = 1MW_g_in

1MW_ww_out=1MW_ww_in

- Önce 2050 orijinal fileini run et sonuclari senin file ile karsilastir ama CO" kisitlamasini yazarak
-
- + Bunun sonucunu sonere sordum dedi ki verimi düsük oldugu icin bu kadar cok cikiyor muhtemelen elektrik üretimi aslinda 0.14 verimle carpinca 22 GW gibi
- Simdi process ayirma isine girelim

How shoult I define lifetime for other technologies. Not all installed capacities are going to be depreciated at the same time

- Burada özellikle wind ve pv leri 3 ve ya 4 e böl duruma göre 0-10 yil önce build edilenler 10-20 yil önce build edilenler 20-30 yil önce build edilenler.

- Onun disinda gas icin de bunu yapabilirsin
- Bu sekilde yapamadiklarin icin ortalama bi yil araligi verebilirsin
- Ve ya uzun süre kullanilan bi teknoloji ise hic yil araligi vermezsin

15.04

- Simdi processleri düzenliyorum
- Gas tibune ekle -----
    - Dummy gas gt
    - Dummy h2 gt
- GT ve CCGT farkli efficiency ve pricelari var
- Gtyi de birden fayla parcaya bölebilirsin
- ~~Renewablelarin hepsini Supim e de eklemen lazim Supim de sadece commoditylei var~~
- Lifetime bi sonraki modelled yila yakin olursa o yamana kadar bi öncekine yakin olursa öncekine kadar tutuyor
- Commodity ratiolari tersine cevir output 1 olacak sekilde
- Supim zaten commodity aliyor oraya hydroyu eklemek lazim sadece
- Gas ve CCGt ayrimina bak
- Kraftwerk listeyi esas kaynak olarak mi alacaksin yoksa Fraunhofer energychartsi mi?
- 2050 dosyasindaki investmen costlari vs düzenle elektricity ratiolarini 1 e cektik orda tam tersi suanda
- Hydro supim bulmak lazim (bulamazsan yilda ortalama kac saat oldugunu bulup saate bölebilirsin

- Simdi BMWi nin Kraftwerklistesi , ve öteki kaynaklarina bak hedef bugün installed capacityleri bulmakl
- In energy charts frauenhofer biomass and biogass are held together
- Also in other generation sources from BMWi but there you can find a column that specifies the type Eingesetzte Biomasse

20.04

- Kaynak listesini Zenginlestirdim
- Installed Capacities icin kaynaklari Smard, Marktdatenstammregister olarak belirledim. Bulamadiklarimi daha eski bmwi datalarindan cikarmayi planliyorum.
- BMWinin senelik raporlarina bakacagim önce
- Sonra 3 senelik

21.04

- Smard datasetinden data cekmek cok zor hem data set net tanimli degil hem de export edemiyorsun. 2005 e kadar olan onshore wind icin 3500 MW gibi bir rakam cikti aslinda bu rakamin 18000 civari olmasi gerekiyor
- Kraftwerkliste'de de suanda aktif GW tutsa da commission yili belirtilmemis cok data var
- Ama BMWinin baska bir kaynagi var orada hangi yil ne kadar kapasite vardi yaziyor. Timeseries. Tabi bu sayinin icinde kac MW kadar tirbünler var PV de cati ve tarla birarada mi yazmiyor ama genel olarak rakamlar bugünün rakamlari ile uyusuyor
- Son olarak marktstammdata cok daha kapsamli. Export etmesi biraz tahmetli ama en kücük unite kadar registered ve aslinda bu kaynagin degerleri timeseries datadan daha fazla cikiyor.

- Ama exportu cok zahmetli oldugundan renewableslar icin timeseries datasini kullaniyorum
- Orada en son 2019 datasi var BMWiden 2020 toplam kapasitesi ile oradaki 2019 arasindaki farki ekliyorum

- Hydropoweri da 3 ayri gruba böldüm cunku depp 50 yil ve 5500MW in 4000 i 1990 öncesinden kalma
- Burada net bi weri olmadigi icin sanki 2030 da 2000MW 2040 da 2000 MW 2050de 800 MW tekrar invest edilmesi gerekiyormus gibi modelledim.
- Ama kapacity upi sabit almayi planliyorum

What is levelized cost of energy:

The levelized cost of electricity (LCOE) in electrical energy production can be defined as the present value of the price of the produced electrical energy (usually expressed in units of cents per kilowatt hour), considering the economic life of the plant and the costs incurred in the construction, operation and maintenance, and the fuel costs.

Aus <https://www.sciencedirect.com/topics/engineering/levelized-cost-of-electricity>

$$\mathrm{LCOE_{wind}} = \frac{\sum_{t=1}^{n} (I_t + O\&M_t - \mathrm{PTC}_t - D_t + T_t + R_t)\frac{1}{(1+i)^t}}{\mathrm{IF}\sum_{t=0}^{t=n-1} P_t} \qquad (25.1)$$

where

$\mathrm{LCOE_{wind}}$ = Generation $\left[\text{cost cents (kW h)}^{-1}\right]$
I_t = Investment made in year t [$]
$O \& M_t$ = Operation and maintenance in year t [$]
PTC_t = Production tax credit [$]
D_t = Depreciation credit [$]
T_t = Tax levy [$]
R_t = Royalties or land rents [$]
IF = Intermittence factor
P_t = Electrical generation capacity in year t [kW h]
G_t = Electrical energy generation in year t [kW h], G_t = IF
n = Duration of the generation period [years]
i = Discount rate

How to Calculate the LCOE?

The LCOE can be calculated by first taking the net present value of the total cost of building and operating the power generating asset. This number is then divided by the total electricity generation over its lifetime. The total costs associated with the project generally will include:

- The initial cost of investment expenditures (I)
- Maintenance and operations expenditures (M)
- Fuel expenditures (if applicable) (F)

The total output of the power-generating asset will include:

- The sum of all electricity generated (E)

The last two important factors to be considered in the equation are:

- The discount rate of the project (r)
- The life of the system (n)

Aus <https://corporatefinanceinstitute.com/resources/knowledge/finance/levelized-cost-of-energy-lcoe/>

$$\mathrm{LCOE} = \frac{\text{NPV of Total Costs Over Lifetime}}{\text{NPV of Electrical Energy Produced Over Lifetime}}$$

$$\mathrm{LCOE} = \frac{\sum \frac{(I_t + M_t + F_t)}{(1+r)^t}}{\sum \frac{E_t}{(1+r)^t}}$$

	PV rooftop small (5-15 kWp)	PV rooftop large (100-1000 kWp)	PV utility-scale (> 2 MWp)	Wind onshore	Wind offshore	Biogas	Brown coal	Hard coal	CCGT	GT
Lifetime [in years]	25	25	25	25	25	30	40	40	30	30
Share of debt	80%	80%	80%	80%	70%	80%	60%	60%	60%	60%
Share of equity	20%	20%	20%	20%	30%	20%	40%	40%	40%	40%
Interest rate on debt	3.5%	3.5%	3.5%	4.0%	5.5%	4.0%	5.5%	5.5%	5.5%	5.5%
Return on equity	5.0%	6.5%	6.5%	7.0%	10.0%	8.0%	11.0%	11.0%	10.0%	10.0%
WACC nominal	3.8%	4.1%	4.1%	4.6%	6.9%	4.8%	7.7%	7.7%	7.3%	7.3%
WACC real	1.8%	2.1%	2.1%	2.5%	4.8%	2.7%	5.6%	5.6%	5.2%	5.2%
OPEX fix [EUR/kW]	2.5% of CAPEX	2.5% of CAPEX	2.5% of CAPEX	30	100	4.0% of CAPEX	36	32	22	20
OPEX var [EUR/kWh]	0	0	0	0.005	0.005	0	0.005	0.005	0.004	0.003
Degradation	0.0025	0.0025	0.0025	0	0	0	0	0	0	0

Table 2: Input parameter for LCOE calculation. The real WACC is calculated with an inflation rate of 2%.

From

Yarin:
Yeni buldugun sourcedan capacityleri proovla 8 in ren cap
GT ve CCGT ile ilgili düsün
KWK ile ilgili de düsün

----- bdewden kapasiteleri al +
------ Kraftwerklisteden eklemedigimiz bir kapasite var mi bak
------- GT CCGT ve KWK ile ilgili düsün

- Biogas 20 is modelled because it has 30 y lifetime and 3500 MW was already installed in 2010 [Zeitrehe]