SINTEF

Optimal scheduling of hybrid power plants

Bjørnar Fjelldal, Ellen Krohn Aasgård

September 12th 2022

Hybrid power plants

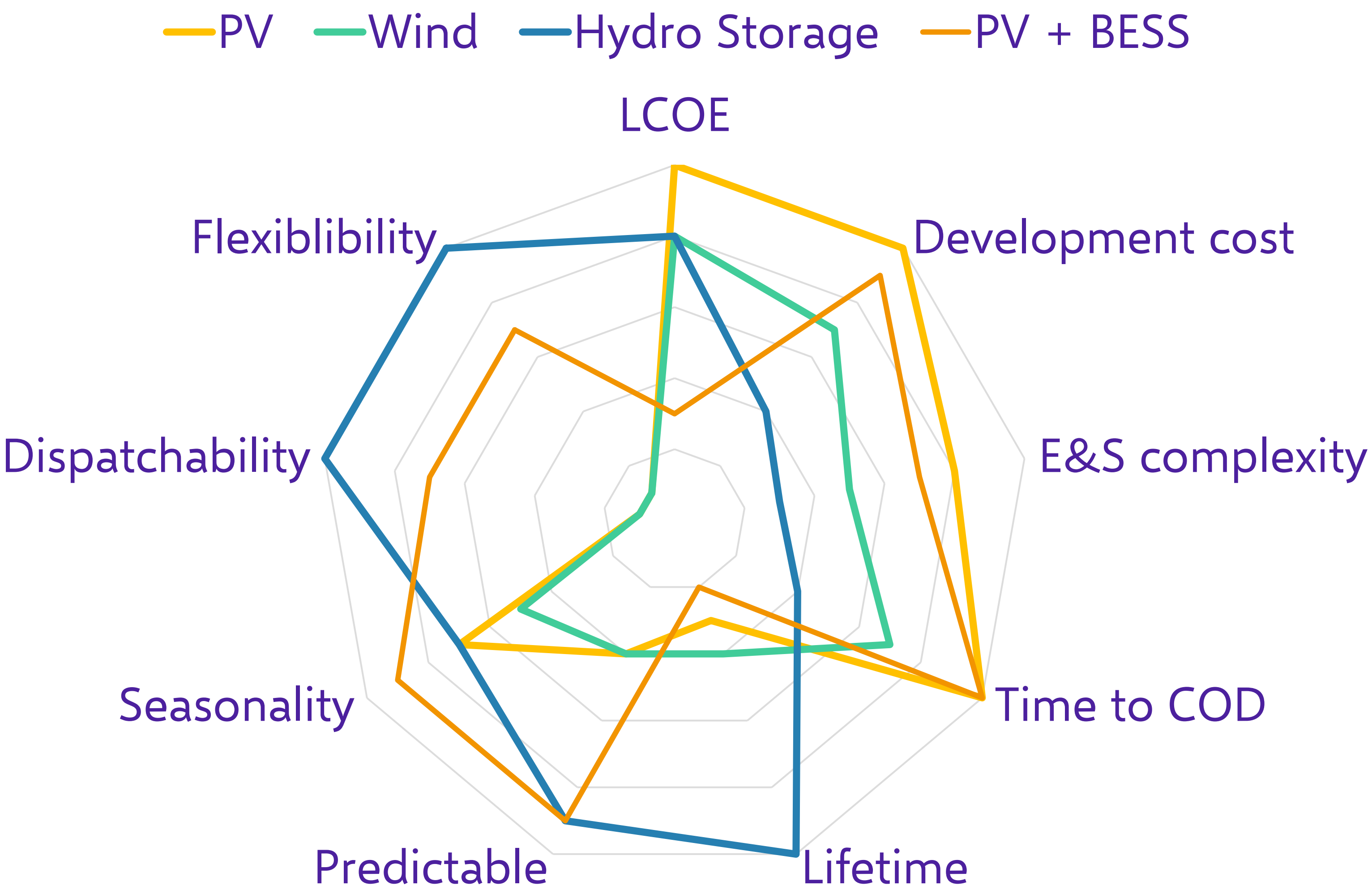
- Future energy supply will be based on renewable technologies: wind, solar and hydro…
- Solar and wind are becoming cost-competitive, but lack important capabilities in terms of storage, reliability and supplying ancillary services
- By combining the cheap energy from solar and/or wind with the regulating capabilities of hydropower, hybrid power plants have a vast potential for supplying affordable, secure and robust energy globally.

7 AFFORDABLE AND CLEAN ENERGY

Scatec Multiconsult

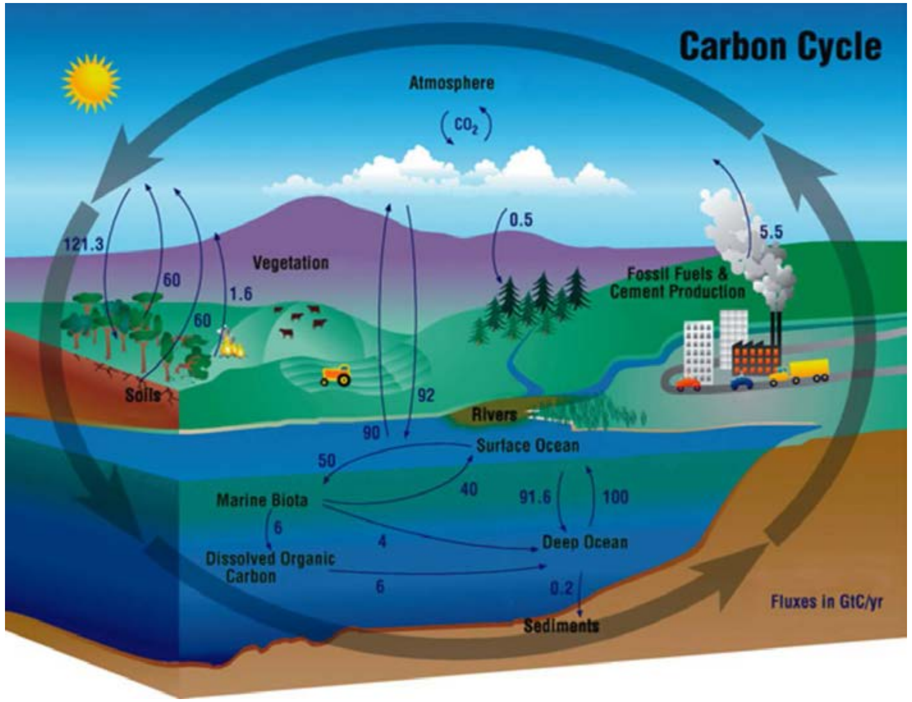
IFE SINTEF prediktor Ocean Sun

Comparing characteristics of renewables



Floating-PV and hydro hybrid power plants

- Efficient use of land areas and resources
  - Surface area of hydro reservoir is "free"?
  - Common grid connection point
  - Inflow and irradiation - correlation and seasonality?
- Floating-PV limits evaporation from hydro reservoirs
  - But how will it affect other environmental/eco-system/biology aspects of the reservoirs and rivers?
- Constantly balancing PV will lead to more wear on hydro equipment and rapid changes in water flows
  - Battery storage may help
- Hybridization for single systems or through the market?
  - Each producer balancing their own portfolio may lead to lower liquidity in the market?
  - But very good in areas where the power market/grid is still under development



Typical scheduling challenges

For hydro alone:

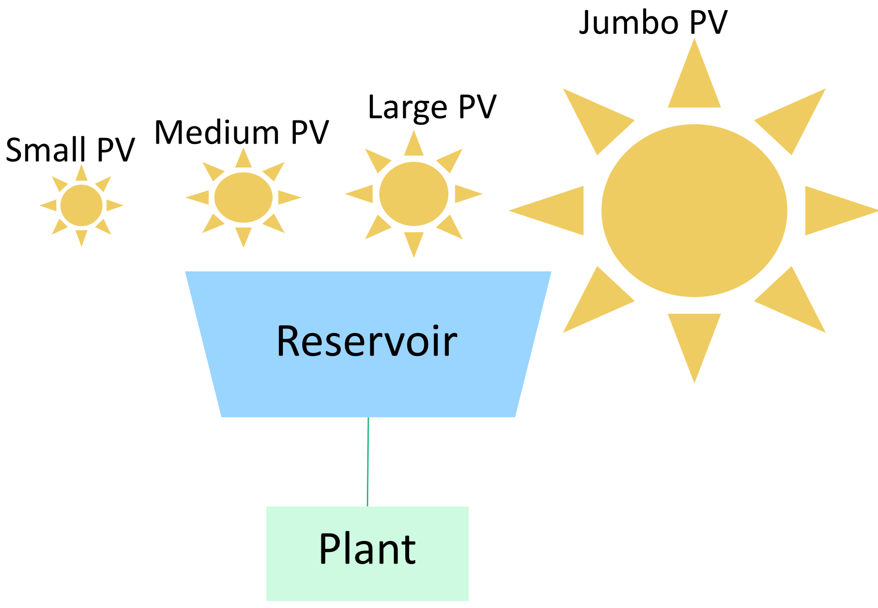
- Complex hydro systems with varying sized reservoirs in series/parallel or other complex configurations
- Complex hydraulic couplings, flow patterns in rivers and tunnels
- Wear, costs and reduced lifetime of equipment due to balancing/flexible operations
- Tight environmental constraints in cascaded systems – minimum flows, ramping, reservoir level, state-dependent constraints…
- Complex market commitments – energy, balancing, reserves…
- Uncertainty in inflow and prices, forecasting accuracy, historical data and models vs climate change

For hybrid plants:

- All of the above!
- Uncertainty for solar (both seasonal, short-term and very-short term) and also joint/correlated uncertainty for solar/inflow/prices/load
- Short-term variability of solar necessities more frequent re-optimizations, .ie. Larger potential for autonomous scheduling

Hypothesis: Large PV capacity compared to hydro capacity will amplify the problems?

Example cases



Jumbo PV

Small PV

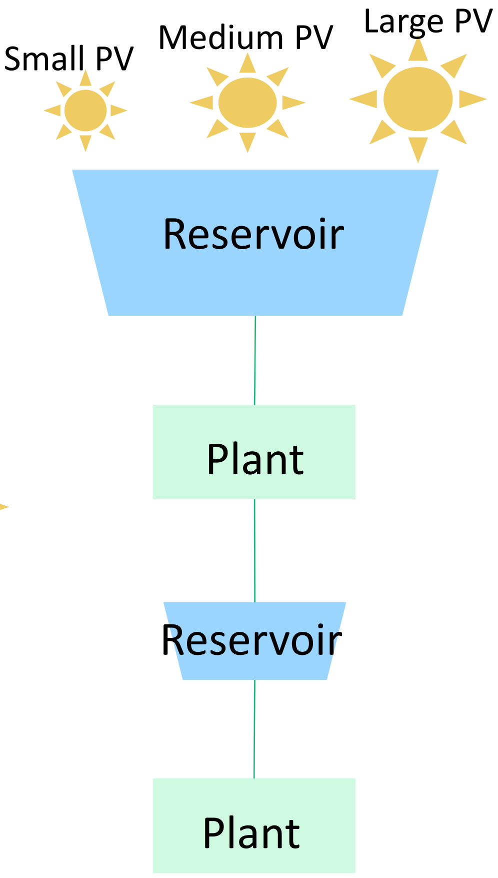
Medium PV

Large PV

Reservoir

Plant

Simple



Small PV

Medium PV

Large PV

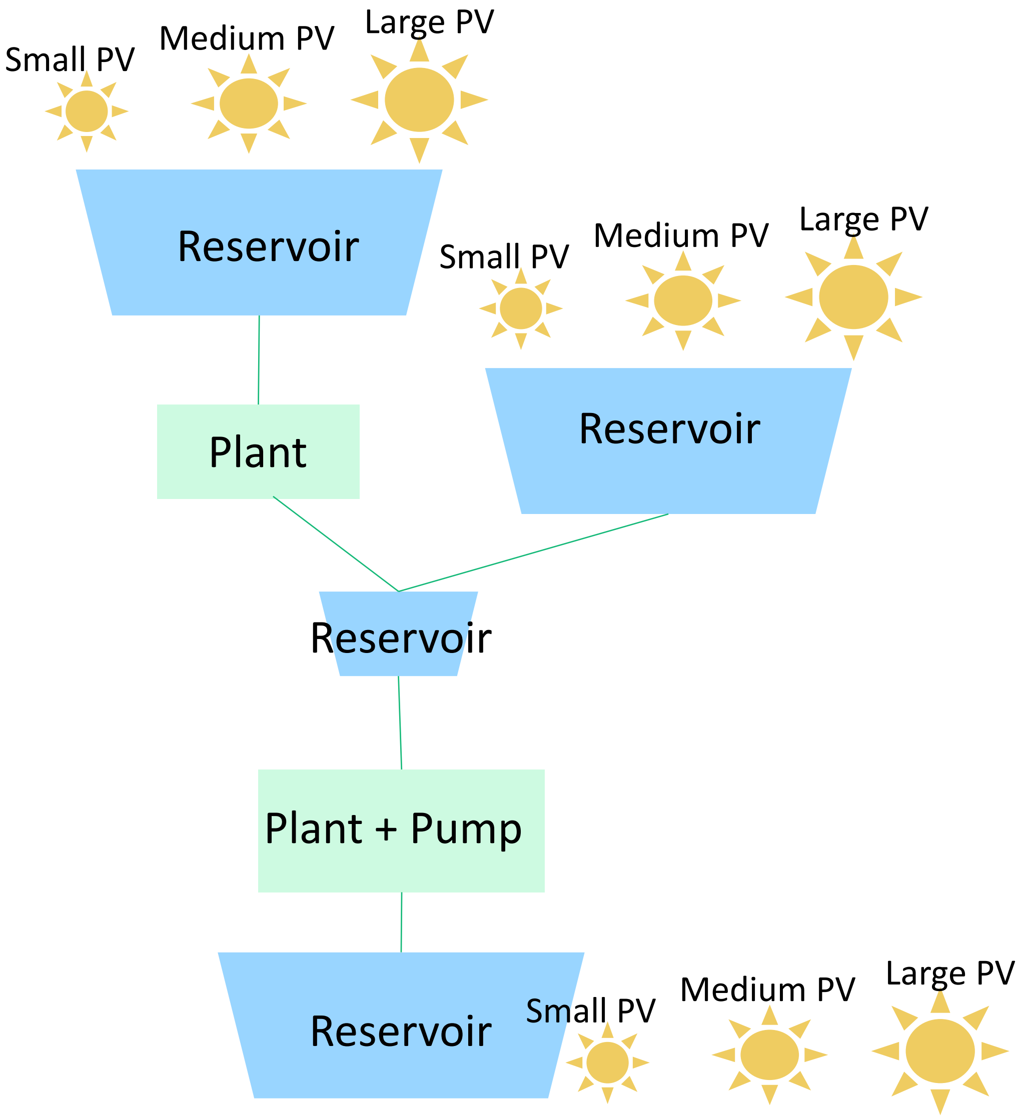
Reservoir

Plant

Reservoir

Plant

Cascade



Small PV

Medium PV

Large PV

Reservoir

Small PV

Medium PV

Large PV

Plant

Reservoir

Reservoir

Plant + Pump

Reservoir

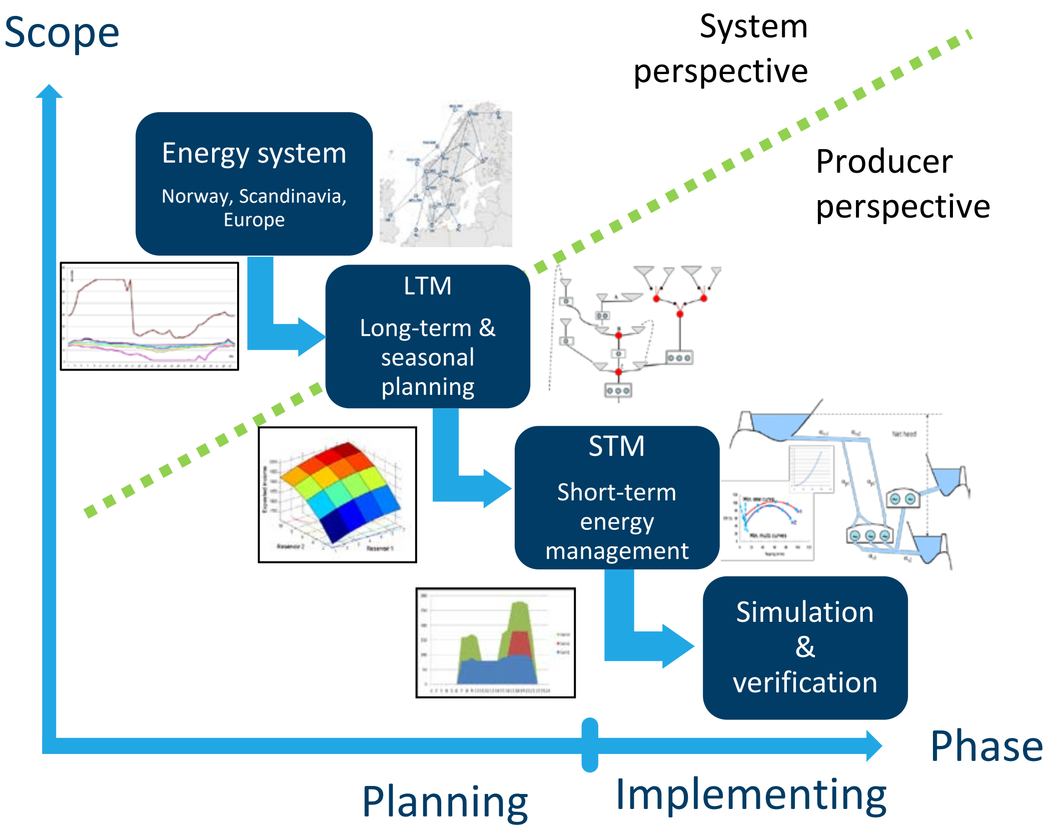
Small PV

Medium PV

Large PV

Complex cascade + pump

SINTEF's hydropower models



Scope

System perspective

Producer perspective

Energy system

Norway, Scandinavia, Europe

LTM

Long-term & seasonal planning

STM

Short-term energy management

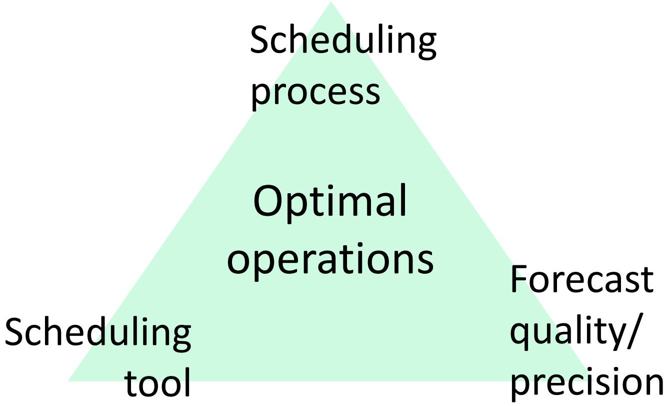
Simulation & verification

Phase

Planning

Implementing

- We have set up analysis for hybrid plants using SINTEF's hydropower models
  - STM: SHOP
  - LTM/seasonal: ProdRisk
- Modelled PV as a "solar market" where energy can be bought at zero cost
- Solar market + hydro production = cover market load



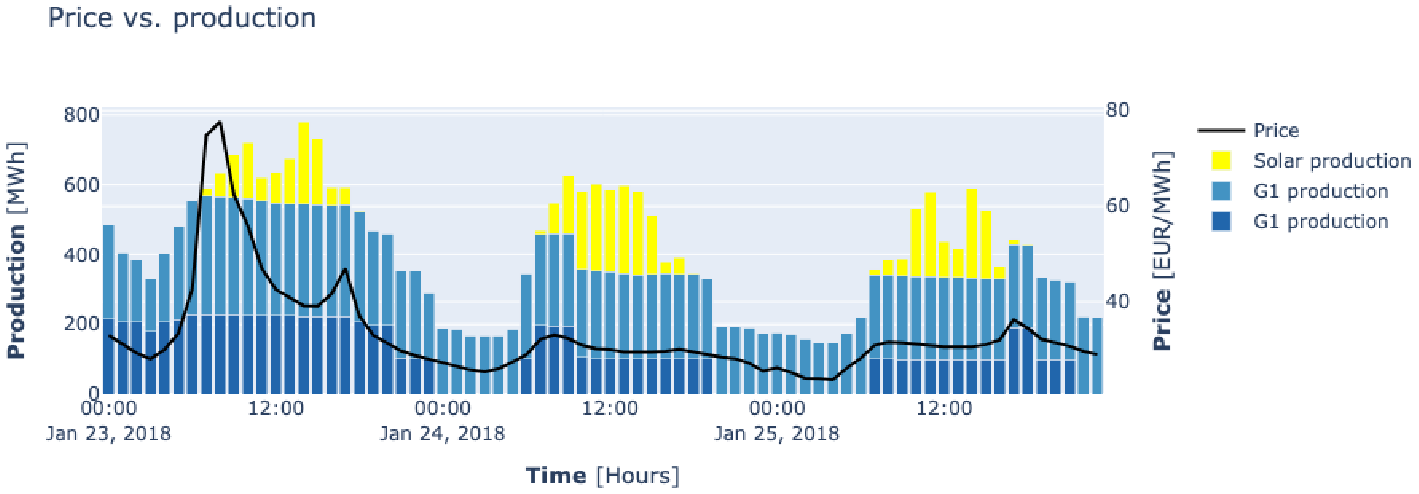
Scheduling process

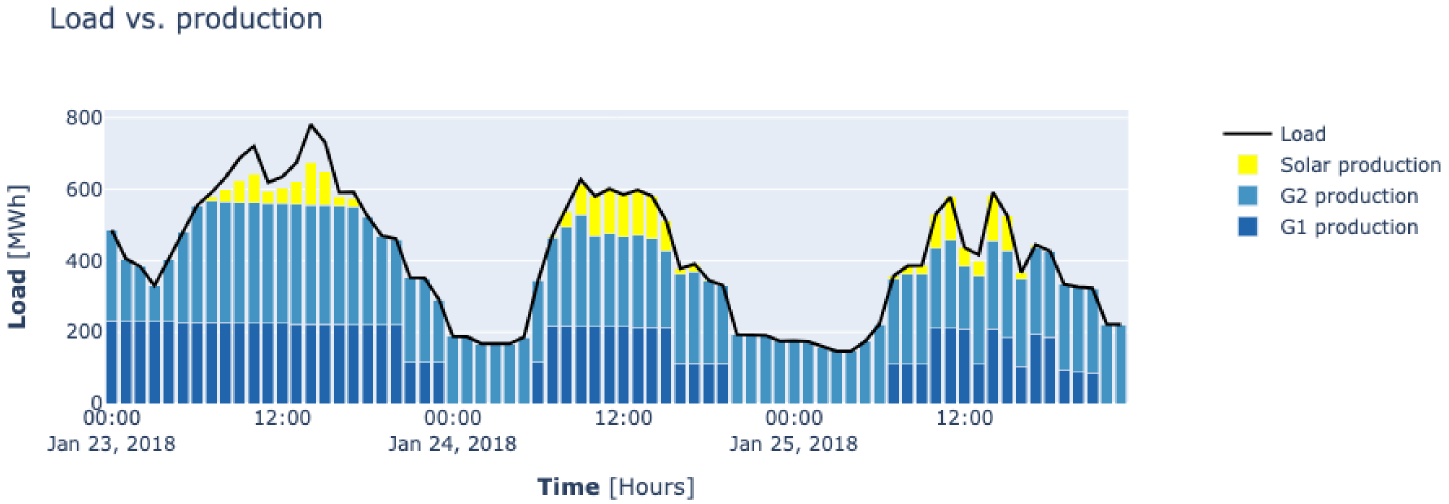
Optimal operations

Scheduling tool

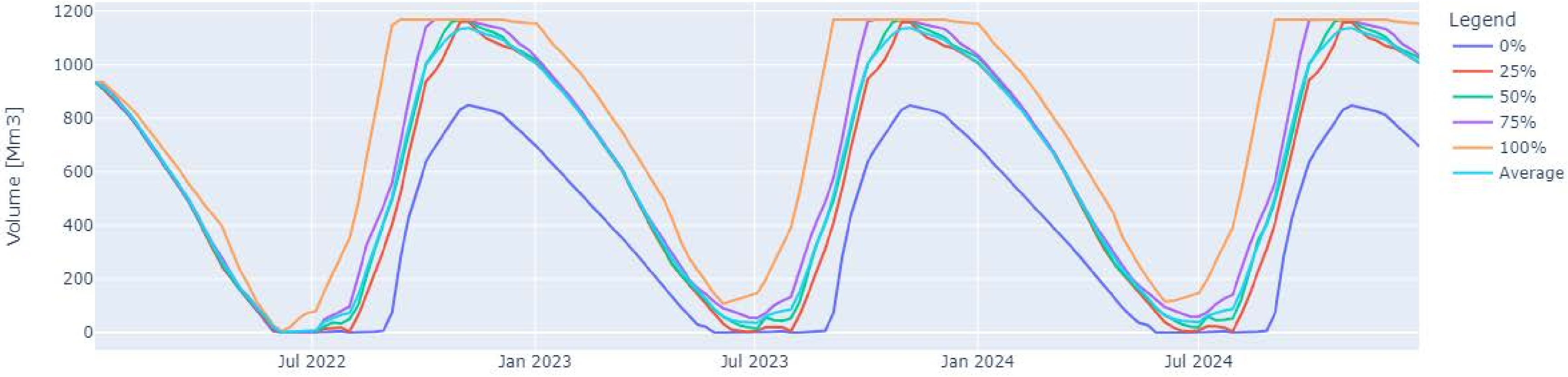
Forecast quality/ precision

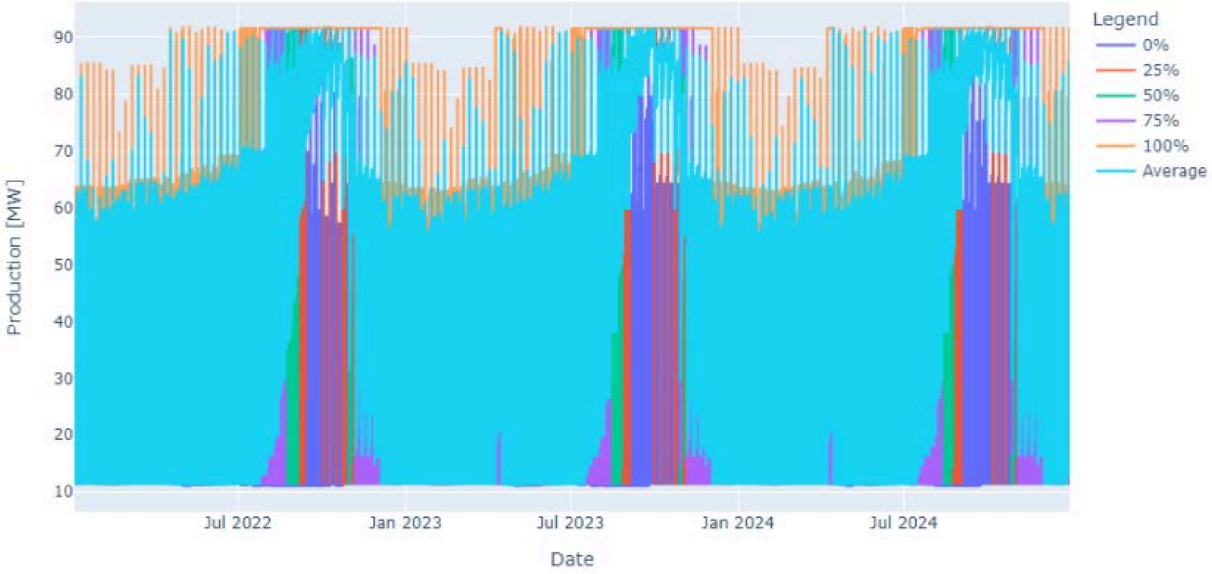
Initial POC for short-term scheduling (SHOP)





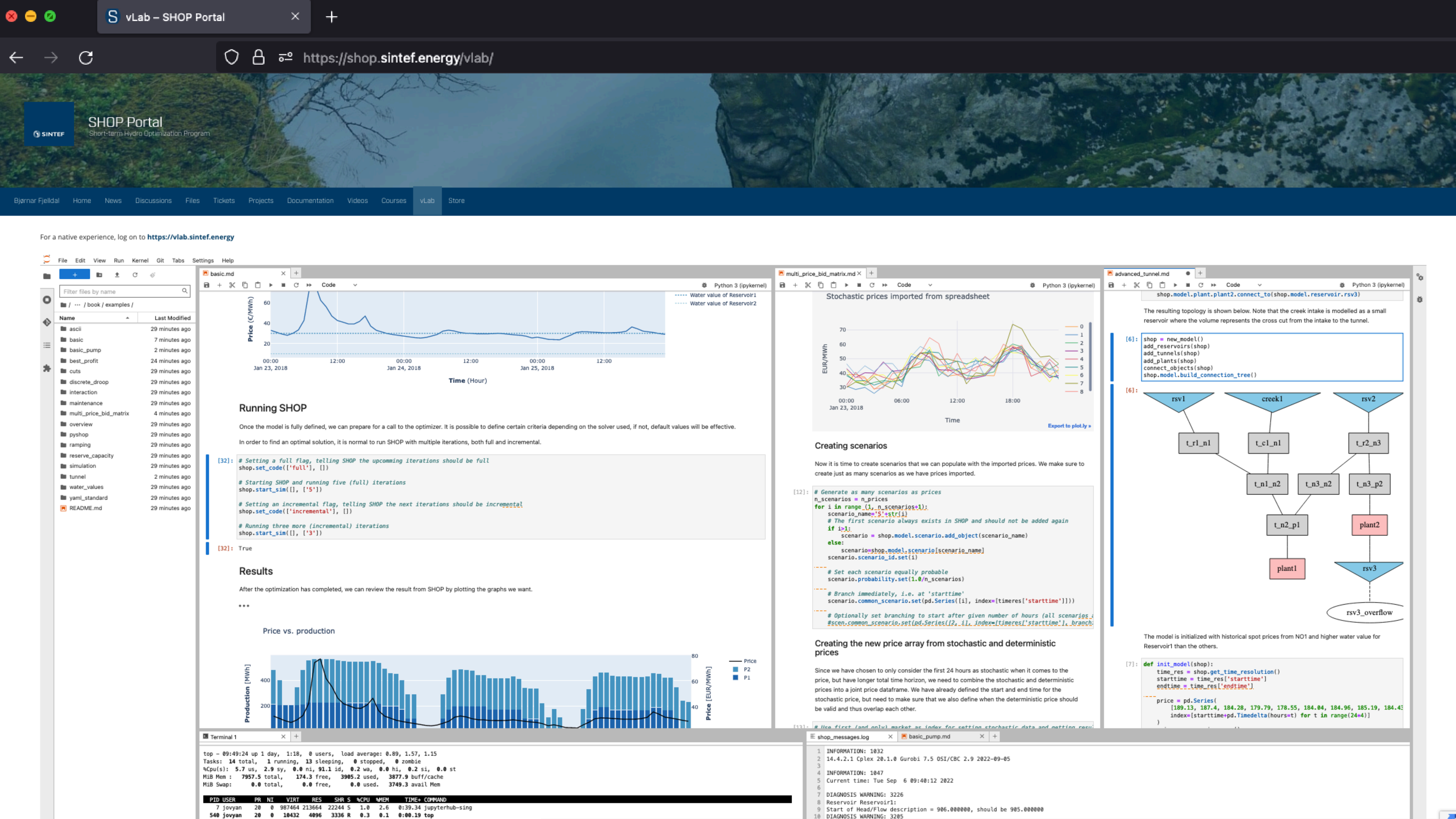
Initial POC for seasonal scheduling (ProdRisk)





Technology for a better society

Credit: Hans Olaf Hågenvik



For a native experience, log on to **https://vlab.sintef.energy**

Running SHOP

Once the model is fully defined, we can prepare for a call to the optimizer. It is possible to define certain criteria depending on the solver used, if not, default values will be effective.

In order to find an optimal solution, it is normal to run SHOP with multiple iterations, both full and incremental.

[32]:

```
# Setting a full flag, telling SHOP the upcomming iterations should be full
shop.set_code(['full'], [])

# Starting SHOP and running five (full) iterations
shop.start_sim([], ['5'])

# Setting an incremental flag, telling SHOP the next iterations should be incremental
shop.set_code(['incremental'], [])

# Running three more (incremental) iterations
shop.start_sim([], ['3'])
```

[32]:

```
True
```

Results

After the optimization has completed, we can review the result from SHOP by plotting the graphs we want.

Creating scenarios

Now it is time to create scenarios that we can populate with the imported prices. We make sure to create just as many scenarios as we have prices imported.

[12]:

```
# Generate as many scenarios as prices
n_scenarios = n_prices
for i in range (1, n_scenarios+1):
    scenario_name='S'+str(i)
    # The first scenario always exists in SHOP and should not be added again
    if i>1:
        scenario = shop.model.scenario.add_object(scenario_name)
    else:
        scenario=shop.model.scenario[scenario_name]
    scenario.scenario_id.set(i)

    # Set each scenario equally probable
    scenario.probability.set(1.0/n_scenarios)

    # Branch immediately, i.e. at 'starttime'
    scenario.common_scenario.set(pd.Series([i], index=[timeres['starttime']]))

    # Optionally set branching to start after given number of hours (all scenarios a
    #scen.common_scenario.set(pd.Series([2, i], index=[timeres['starttime'], branch
```

Creating the new price array from stochastic and deterministic prices

Since we have chosen to only consider the first 24 hours as stochastic when it comes to the price, but have longer total time horizon, we need to combine the stochastic and deterministic prices into a joint price dataframe. We have already defined the start and end time for the stochastic price, but need to make sure that we also define when the deterministic price should be valid and thus overlap each other.

[13]:

```
# Use first (and only) market as index for setting stochastic data and getting resu
```

```
    shop.model.plant.plant2.connect_to(shop.model.reservoir.rsv3)
```

The resulting topology is shown below. Note that the creek intake is modelled as a small reservoir where the volume represents the cross cut from the intake to the tunnel.

[6]:

```
shop = new_model()
add_reservoirs(shop)
add_tunnels(shop)
add_plants(shop)
connect_objects(shop)
shop.model.build_connection_tree()
```

[6]:

The model is initialized with historical spot prices from NO1 and higher water value for Reservoir1 than the others.

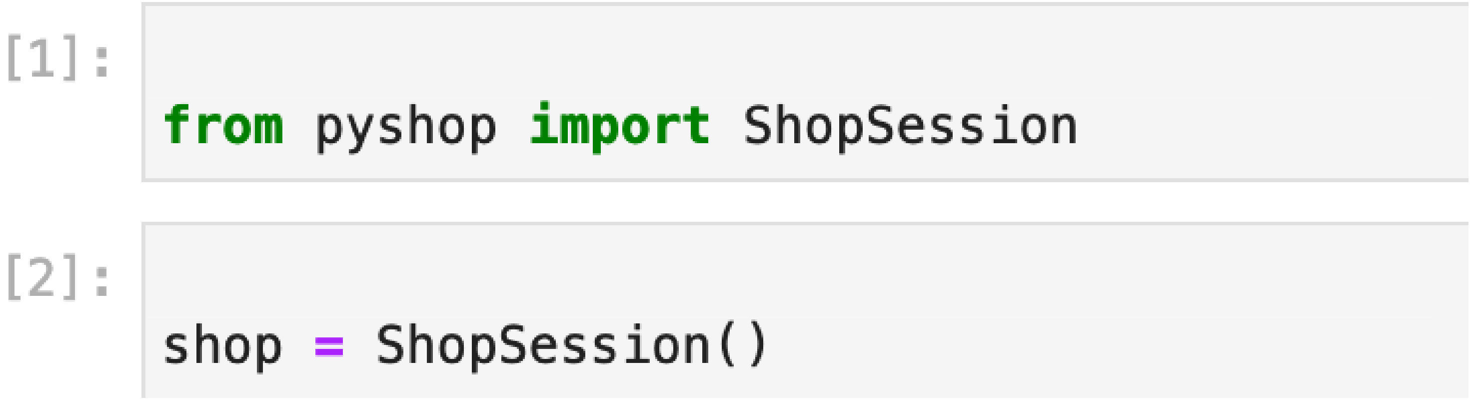
[7]:

```
def init_model(shop):
    time_res = shop.get_time_resolution()
    starttime = time_res['starttime']
    endtime = time_res['endtime']

    price = pd.Series(
        [189.13, 187.4, 184.28, 179.79, 178.55, 184.04, 184.96, 185.19, 184.43
        index=[starttime+pd.Timedelta(hours=t) for t in range(24*4)]
    )
```

```
top - 09:49:24 up 1 day,  1:18,  0 users,  load average: 0.89, 1.57, 1.15
Tasks:  14 total,   1 running,  13 sleeping,   0 stopped,   0 zombie
%Cpu(s):  5.7 us,  2.9 sy,  0.0 ni, 91.1 id,  0.2 wa,  0.0 hi,  0.2 si,  0.0 st
MiB Mem :   7957.5 total,    174.3 free,   3905.2 used,   3877.9 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.   3749.3 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
    7 jovyan    20   0  987464 213664  22244 S   1.0   2.6   0:39.34 jupyterhub-sing
  540 jovyan    20   0   10432   4096   3336 R   0.3   0.1   0:00.19 top
```

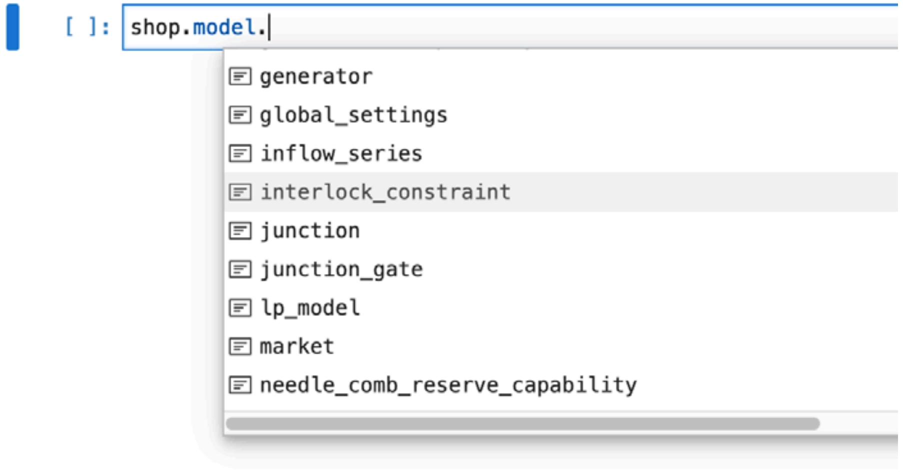
```
INFORMATION: 1032
14.4.2.1 Cplex 20.1.0 Gurobi 7.5 OSI/CBC 2.9 2022-09-05

INFORMATION: 1047
Current time: Tue Sep  6 09:40:12 2022

DIAGNOSIS WARNING: 3226
Reservoir Reservoir1:
Start of Head/Flow description = 906.000000, should be 905.000000
DIAGNOSIS WARNING: 3205
```

More robust and flexible model coupling between SHOP and ProdRisk



```
[1]: from pyshop import ShopSession
```

```
[2]: shop = ShopSession()
```

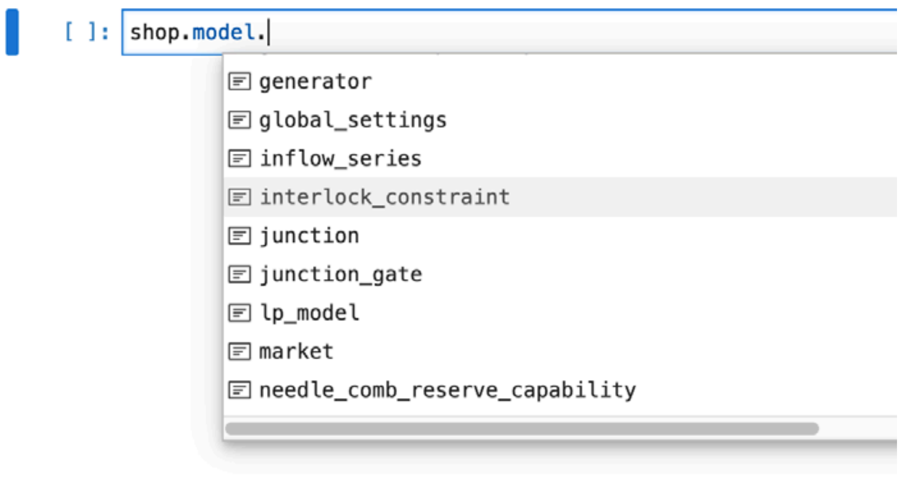


```
[ ]: shop.model.
```

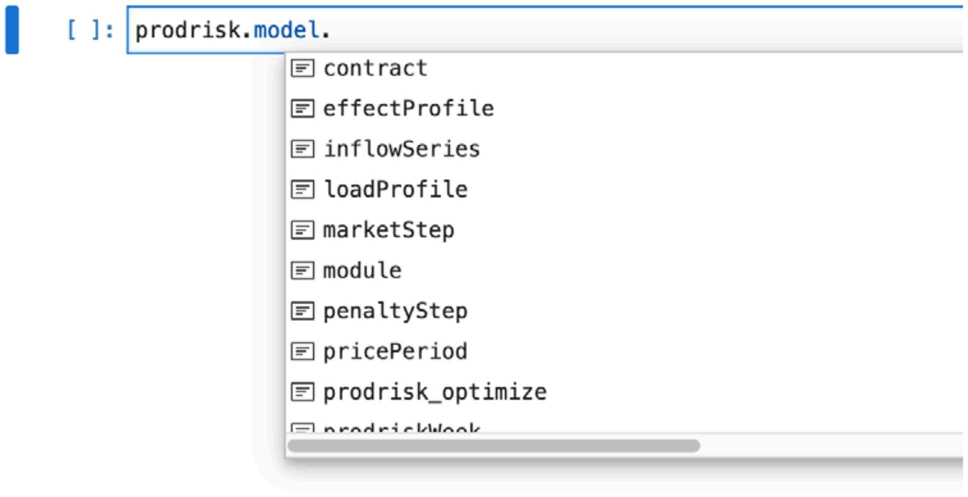
More robust and flexible model coupling between SHOP and ProdRisk

```
[1]: from pyprodrisk import ProdriskSession
     from pyshop import ShopSession
```

```
[2]: prodrisk = ProdriskSession()
     shop = ShopSession()
```



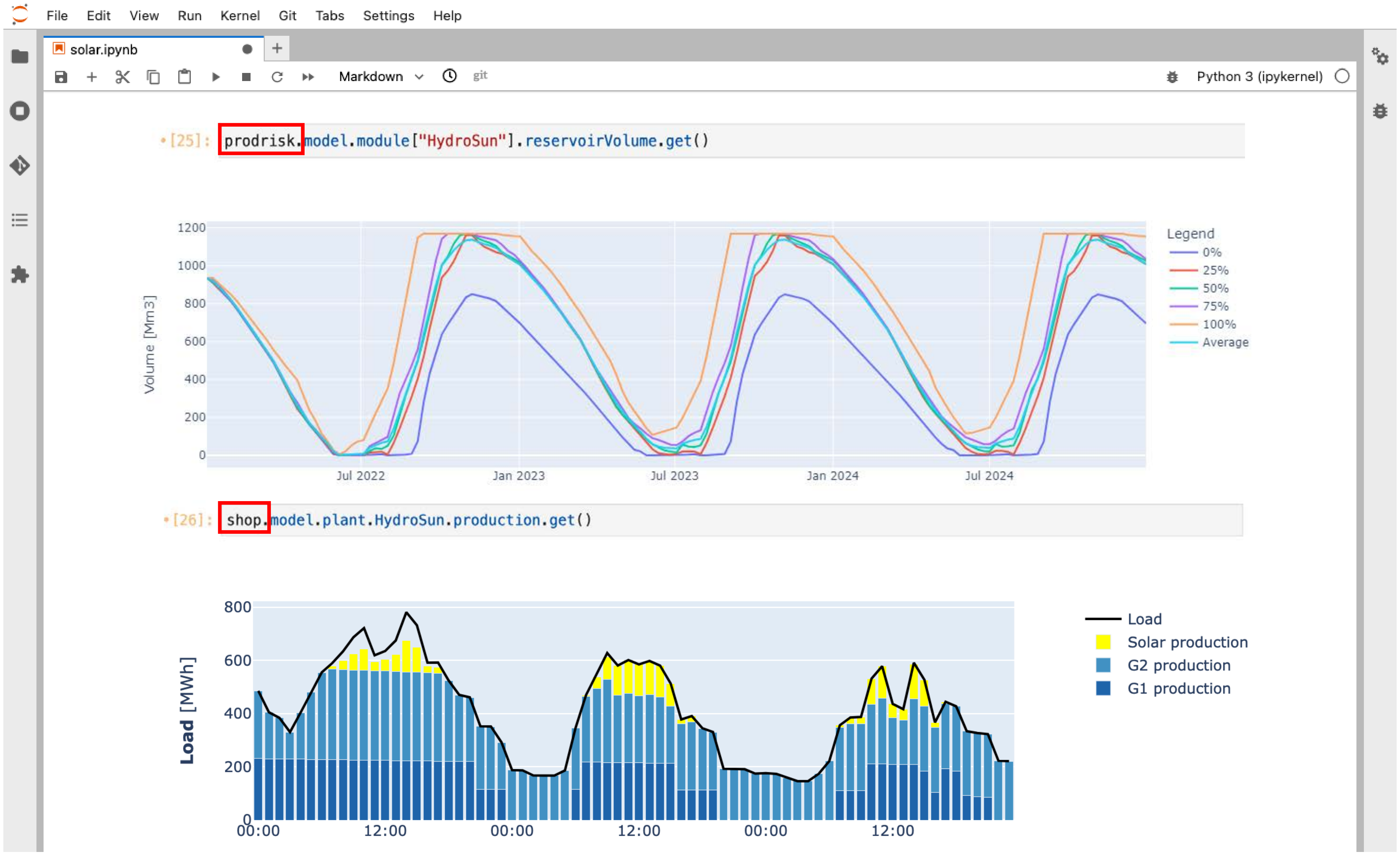
```
[ ]: shop.model.
```



```
[ ]: prodrisk.model.
```

SINTEF

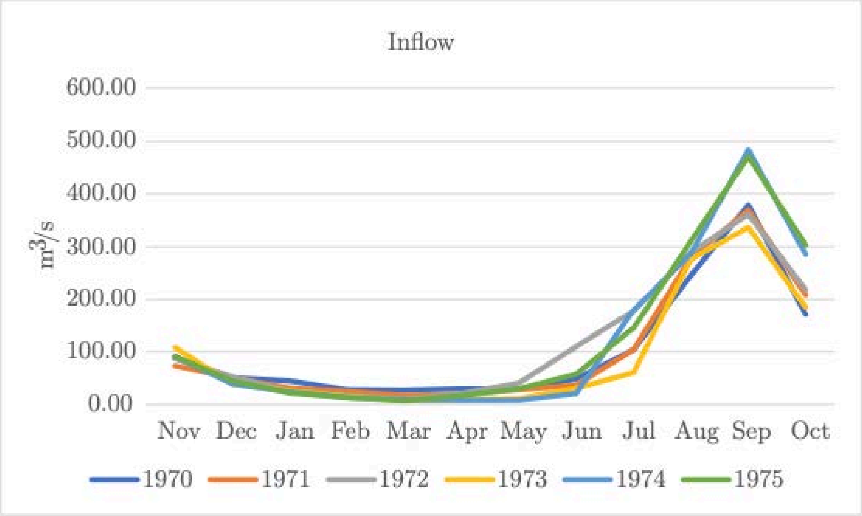
- Same (running) kernel
- Same environment
- Same user interface
- Easier debugging
- Better data handling
- More flexible
- More robust

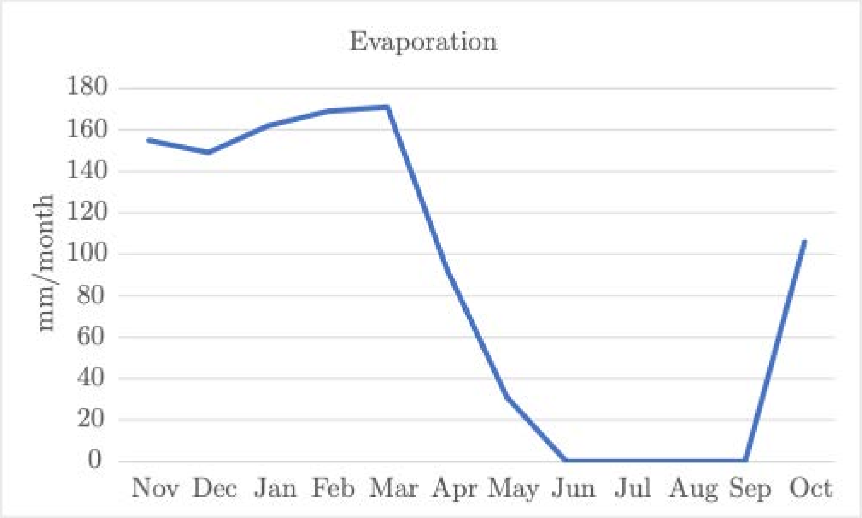


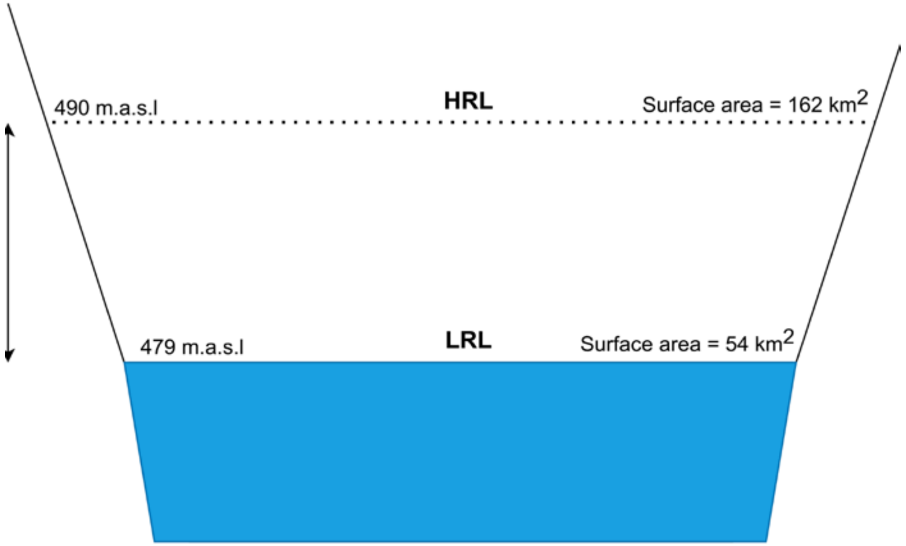
```
[25]: prodrisk.model.module["HydroSun"].reservoirVolume.get()
```

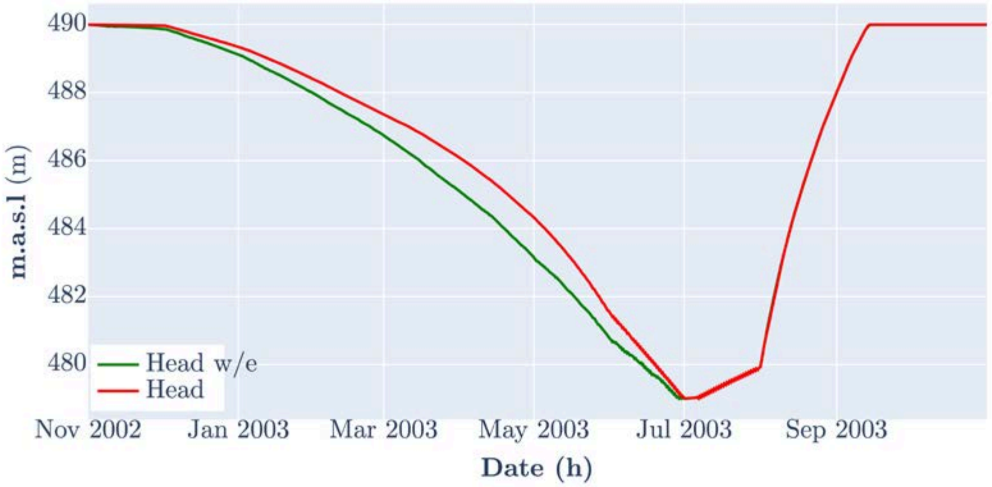
```
[26]: shop.model.plant.HydroSun.production.get()
```

Evaporation









Credit: Benjamin Trondsen

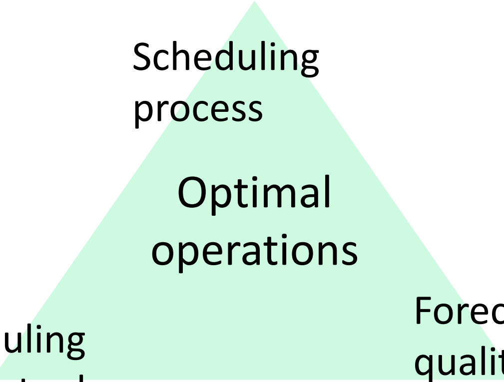
Future work

- Project period: 2021-2024

- Need for scheduling models to be updated more frequently in order to re-plan closer to real-time
- The optimal hydro-PV hybrid scheduling system will realize the complementation of energy generation over seasons, days, hours, and seconds
- Establish a new reservoir trajectory curve that includes inflow, power generation as well as solar generation effect

- PhD + MSc students also involved

7 AFFORDABLE AND CLEAN ENERGY



Scheduling process

Optimal operations

Scheduling tool

Forecast quality/ precision

SINTEF

Technology for a better society

This presentation is based on work by
Jiehong Kong, Hans Ivar Skjelbred, Hans Olaf Hågenvik, Benjamin Trondsen, Bjørnar Fjelldal and Ellen Krohn Aasgård