Technical University of Munich
Department of Electrical and Computer Engineering
Institute for Renewable and Sustainable Energy Systems

Bachelor's Thesis

Demand Side Management in Bavaria Implementation and Application

written by

Bekir Okan Akca

Matr.Nr. 03636585

Submitted to the	Institute for Renewable and Sustainable Energy Systems Technical University of Munich
Head	Prof. Dr. rer. nat. Thomas Hamacher
Advisor	Magdalena Dorfner, M.Sc.

Abstract

The growing energy needs of the World caused more research in the field of energy. Correspondingly, DSM (demand-side management) gained more attention and found valuable to research on. Even though previous models of DSM had problems on real-world applicability, the latest formulation by Zerrahn-Schill is increased the real-world relevance and shown promising results. In this thesis, the formulation by Zerrahn-Schill, which overcame the problems of Göransson et al., is explained. Afterwards it is used in a power-system model, called urbs. The outcome of the implementation of Zerrahn-Schill formulation to urbs is evaluated. As a result, the implementation and application of the Zerrahn-Schill formulation is shown, that urbs with DSM capabilities is more profitable and realistic, on the other hand, it needs more memory to run and takes longer to find a solution for given input data.

Statement of Academic Integrity

I,

Last name: Akca
First name: Bekir Okan
ID No.: 03636585

hereby confirm that the attached thesis,

Demand Side Management in Bavaria
Implementation and Application

was written independently by me without the use of any sources or aids beyond those cited, and all passages and ideas taken from other sources are indicated in the text and given the corresponding citation. Tools provided by the institute and its staff, such as models or programs, are also listed. These tools are property of the institute or of the individual staff member. I will not use them for any work beyond the attached thesis or make them available to third parties.

I agree to the further use of my work and its results (including programs produced and methods used) for research and instructional purposes.

I have not previously submitted this thesis for academic credit.

Munich, February, 18, 2016 ____________________

Contents

Chapter 1

Introduction

Energy is one of the essential factors to the existence of mankind. With growing population on earth, energy needs are increasing and leading to a global energy crisis. There are tons of thousands researches, which is done or going on, in order to find a solution to the World's growing energy crisis. The non-renewable energy sources are running out and they will not last more than a century [1]. This dilemma points people to look for options for the sake of preventing this crisis.

1.1 DSM Examples in the World

The term DSM (Demand-Side Management) is told firstly following the time of the 1973 and 1979 energy crisis [2]. After that, the application of DSM gained attention in many countries just as Zerrahn-Schill in their article mentioned[3]. Nowadays, DSM technologies become more important and are implemented in many power-system models. For instance, the government of the state Queensland, Australia plans to integrate some devices in certain households such as air conditioners, pool pumps, and hot water systems. With these devices energy companies can cycle the usage remotely during peak hours. According to [4], their plan includes giving financial incentives to customers who use electricity during off-peak hours. Moreover, in Toronto, Canada the main energy distributor Toronto Hydro had over 40.000 people to signed up to have remote devices attached their air conditioners working with same principle just as Bradburry mentions [5].

In the State California, USA there are already several DSM programs since 2008 and the California Public Utilities Commission released also a proposed decision in 2015 to improve 2008's DSM programs. In the article of St. John [6], it is mentioned that California sets up for a radical grid-edge integration to increase efficiency with balancing the utility-customer benefits.

1.2 Motivation & Scope of Thesis

DSM has many potentials preventing the growing need for energy in the World. Just as above mentioned, many countries improving their energy grid to improve energy efficiency and to reduce the total cost with several DSM technologies. Achieving this goal can be done with implementing DSM not only to regional energy providers, but also to consumer households. Additionally, DSM decreases spent money on energy with just programming. So that the consumers can profit more with spending less for same energy demand. DSM helps also integrating renewable energy sources. This thesis is written because of these mentioned reasons.

The aims of this thesis are describing the concepts of DSM, implementing a new formulation into the model urbs [7] and application of DSM capabilities to the Bavarian Model. Moreover, in section 2 formulations of Göransson et al. [8] and Zerrahn-Schill [3] are explained with their advantages and disadvantages. The mathematical documentation of the formulations can be also found in that section.

Afterwards, the brief introduction to the model urbs is made and the working principles of urbs is clarified in section 3. The implementation process of DSM into the urbs is also with examples demonstrated and some of the modified code can be seen. Additionally, the runtime comparison of urbs after adding the DSM capabilities is shown in this section.

After several runs of the DSM capable urbs, the results are gathered and evaluated. In section 4, the gathered results are shown with taken outcomes of urbs with Bavarian Model. Finally, in section 5 the thesis is concluded.

Chapter 2

Demand-Side Management

Without having a common definition of Demand-Side Management (DSM), it refers as generally how to respond to the real time prices [9] [10]. There are several aspects of DSM and this thesis is mainly focused at *load shifting*.

The working principles of DSM is simple; storing energy when the prices are low and using the stored energy, when the prices are high [11]. There are several requirements to be able to implement the DSM capabilities on a system. First, the demand needs to be known as certain. Secondly, the system is needed a battery cell in order to store energy and use it at different times. Of course, it seems like a big investment to purchase a battery cell, which can store lots of energy, however as a long time investment, it should compensate the spent money with the gained profit of DSM. It can also have a wide range of usage; such as at home or at big enterprises like factories, where the energy usage is crucial point for the sake of gaining more profit.

DSM also helps to integrate renewable energy sources [12]; such as bio-energy, wind-energy, solar-energy, etc. These are some of the most important renewable energy sources in Germany and as a country, Germany tries to boost clean energy usage with lots of research and laws such as "The Renewable Energy Act [13]". For a better future, DSM can and will provide boosting usage of renewable energy sources. The effect of DSM would be of course less than for example shutting down all of the nuclear plants, however the effect could be improved with using DSM all over the country.

2.1 Load Shifting

Load Shifting is one of the aspects of DSM. It is also the main point of this thesis. In basic words, load shifting refers to move the high demand to the off-peak times. The rationale behind load shifting is smoothing the demand line, so that the energy sources can satisfy the demand without overworking. It can also increase the usage of renewable sources [14].

It is shown in the figure 2.1, that the demand line with DSM capabilities is more flat and smooth than the demand line without DSM. Overall demand is not changed with DSM, but as hourly there are differences. It can be seen, that the hours with high demand are now reduced and the hours with low demand are increased with

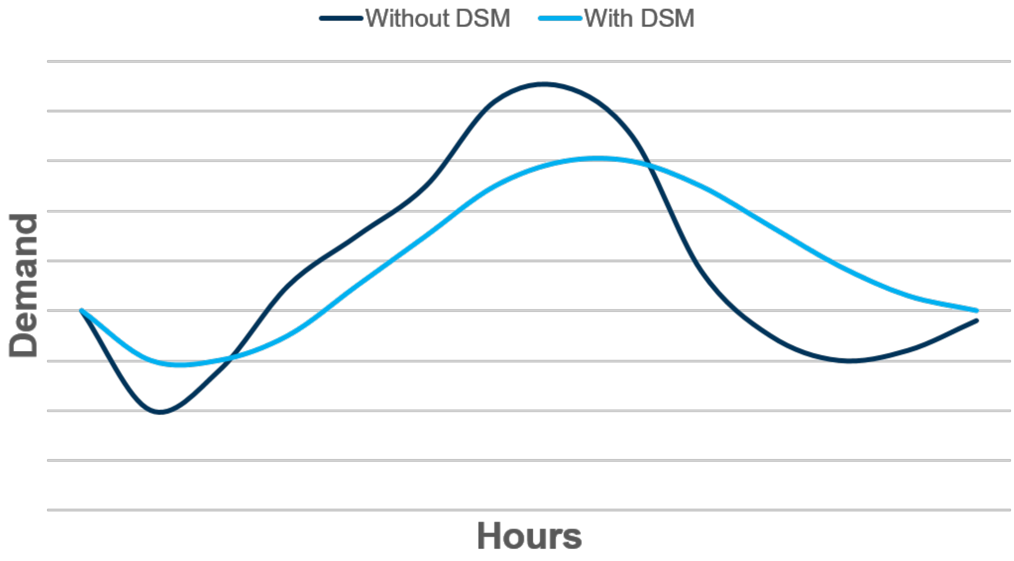


Figure 2.1 Demand Smoothing

DSM. According the changing prices of energy, generating power in off-peak times makes sense for money issues.

2.2 Formulations

A formulation within the model is required to run DSM capabilities in any power-system model. There are several formulations of DSM tools to be able to use them in a power-system model. In this thesis Göransson et al. [8] and Zerrahn-Schill [3] Formulation is investigated; how do they work, the problems and also the working principles of them.

The formulations are basically there for setting rules for the variables or constants to make them work in a proper, realistic way. Formulations can also be seen as the rules of the DSM capabilities. Without those rules, variables of DSM could do anything to the demand line, but then of course it would be not pragmatic. In the next sections Göransson et al. [8] and Zerrahn-Schill [3] formulations is explained and a table containing all constants, variable and indexes is in the Appendix included.

2.2.1 Göransson et al. Formulation

So far the formulation, which is introduced by Göransson et al. [8], includes much of the DSM's aspects. It is a concise, linear, and largely convincing formulation and it arises out of six equations, which can be found below.

$$dh_t = \sum_{l=0}^{L-1} dd_{t-l} \qquad \forall t \qquad (2.1)$$

$$dh_t = \sum_{l=1}^{L} ds_{t+l} \qquad \forall t \qquad (2.2)$$

$$dh_t = dh_{t-1} + ds_t - dd_t \qquad \forall t \qquad (2.3)$$

dh_t represents the demand put on hold. Assuming a delay time L, just as in equation (2.1), dh_t needs to be equal to sum of hourly demand delayed dd_t over previous $L-1$ periods, starting from current hour. Likewise; equation (2.2) assures, that dh_t is also equal to sum of hourly demand served ds_t over the next L hours. Demand put on hold is balanced in the equation (2.3) with the previous hour's level of dh_t and the remaining value from demand served minus delayed $(ds_t - dd_t)$

$$dd_t \leq C^{dd} \qquad \forall t \qquad (2.4)$$

$$ds_t \leq C^{ds} \qquad \forall t \qquad (2.5)$$

$$dd_t + ds_t \leq \max\{C^{dd}, C^{ds}\} \qquad \forall t \qquad (2.6)$$

The variables of the formulation should have threshold values in order to work rationally. Equation (2.4) and (2.5) shows, that demand delayed and served have to be less than the threshold capacity C^{dd} and C^{up}. Furthermore, there is one more equation, which is shown in (2.6), prevents the variables to work simultaneously.

Although it is linear and convincing formulation to use on power-system models, there are two critical drawbacks; "undue recovery" and "always starting load shifting with a delay". These problems may cause the model not to get realistic results, and also while load shifting flexibility issues may appear.

Undue recovery means, that the model is using the DSM variables at their full capacity for long hours, which may violate some of the constraints within the model. In the long run undue recovery may cause the power-system models to fail getting realistic results or any result at all. To avoid such problems Zerrahn-Schill [3] formulated a better way to control these DSM variables with adding one more constraint, which force them to work simultaneously and not at full capacity at all times. It will be explained more at following section.

Secondly, their formulation has also flexibility issues. Load shifting starts always with a delay, which means; the model itself cannot increase demand without having a decreased demand at previous hours. In other words, even the first hours in a power-system are off-peak times, where the prices of energy are at their lowest rates, the model cannot decide to purchase more energy and save it to a battery by increasing the demand at that hour. It waits until the peak times to decrease the demand, then the DSM capabilities start to work, which is of course causes some loss on profit in the end.

2.2.2 Zerrahn-Schill Formulation

Formulation, which is introduced by Zerrahn-Schill [3], is designed specifically to overcome the shortcomings of Göransson et al.'s [8] formulation. The problems appeared by Göransson et al. [8] formulation is fixed, while making the power-system model more realistic, flexible and prevented against over-working variables.

There are five equations and within these, two variables and five constants are to be found. The variables DSM_t^{up} and $DSM_{t,tt}^{do}$ represented in Zerrahn-Schill [3] formulation are there for changing the demand. As the name implies, DSM_t^{up} increases; $DSM_{t,tt}^{do}$ decreases the demand. However, just as the equation (2.7) says that the DSM_t^{up} variables have to be compensated by $DSM_{t,tt}^{do}$ within a delay time L. An efficiency factor is there for assuming the compensated demand would be less than the changed, according to real world physics. There are two indexes of $DSM_{t,tt}^{do}$ variables and why it is necessary is explained in the paragraph *Modeled Zerrahn-Schill Formulation*.

$$DSM_t^{up}\eta = \sum_{tt=t-L}^{t+L} DSM_{t,tt}^{do} \qquad \forall t \qquad (2.7)$$

Concerning about the real world physics the variables have to have a threshold capacity value in order to work. In the equations (2.8) and (2.9) are shown, that the variables DSM_t^{up} and $DSM_{t,tt}^{do}$ have the threshold capacities C^{up} and C^{do}. From a realistic approach, setting up about 10% of the average demand is reasonable for the values of these capacities.

$$DSM_t^{up} \leq C^{up} \qquad \forall t \qquad (2.8)$$

$$\sum_{t=tt-L}^{tt+L} DSM_{t,tt}^{do} \leq C^{do} \qquad \forall tt \qquad (2.9)$$

DSM_t^{up} and $DSM_{t,tt}^{do}$ should not work simultaneously. Equation (2.10) prevents it from happening. Because of the equation (2.10) one of the equations (2.8) and (2.9) can be ignored, depending on which one of them is tighter. In the paragraph *Modeled Zerrahn-Schill Formulation*, it is also explained with examples.

$$DSM_{tt}^{up} + \sum_{t=tt-L}^{tt+L} DSM_{t,tt}^{do} \leq \max\{C^{up}, C^{do}\} \qquad \forall tt \qquad (2.10)$$

Equation (2.11) is related to the recovery time of the physical parts of the DSM capabilities. It is not certain, weather physical parts of DSM_t^{up} variables are needed for a cool down time or not. Furthermore, there are differences, when the DSM capabilities are used in different systems. There is no research concerning the recovery time at the moment, therefore recovery time R is set to 1 for every test in this thesis.

$$\sum_{tt=t}^{t+R-1} DSM_{tt}^{up} \leq C^{up}L \qquad \forall t \qquad (2.11)$$

Modeled Zerrahn-Schill Formulation

In contrast of Göransson et al. [8], in Zerrahn-Schill [3] formulation load shifting does not start always with a delay. The reason lies behind the variable $DSM^{do}_{t,tt}$. As mentioned before, $DSM^{do}_{t,tt}$ has two indexes. Therefore these variables can compensate the DSM^{up}_{t} variables, which increased the demand before, after or both.

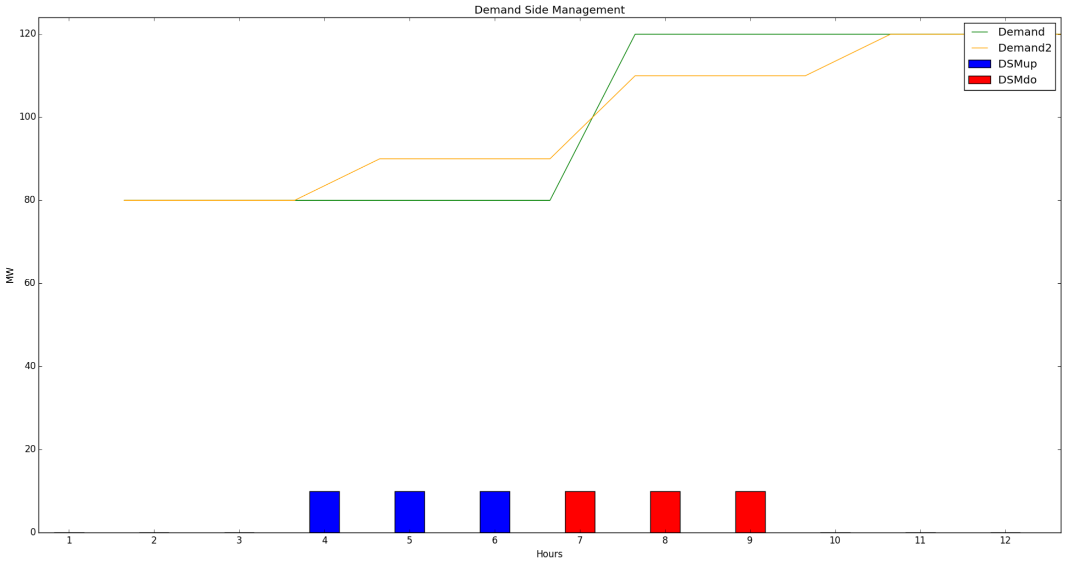


Figure 2.2 Load Shifting; $L = 3$, $\eta = 1$, $R = 1$

In the figure 2.2, twelve hours power-system is realized and demand with DSM is calculated with the equation 2.12:

$$Demand^{DSM}_{t} = Demand_{t} + DSM^{up}_{t} - \sum_{t=tt-L}^{tt+L} DSM^{do}_{t,tt} \quad \forall t \tag{2.12}$$

$DSM^{do}_{t,tt}$ variables in hour tt have to compensate for DSM^{up}_{t} variables in hour t, just as the equation (2.7) shows. It can be seen in the table below, that DSM^{up}_{4}, DSM^{up}_{5} and DSM^{up}_{6} is equal to 10 MWh. Additionally the sum of the $DSM^{do}_{t,tt}$ variables between $DSM^{do}_{4,1}$ to $DSM^{do}_{4,7}$ have to be also equal to 10 MWh to balance DSM^{up}_{4}. Most profitable option would be setting up the $DSM^{do}_{4,7}$ value to 10 MWh, because starting from 7th hour there is a peak time of demand. Likewise; DSM^{up}_{5} and DSM^{up}_{6} have to be balanced by $DSM^{do}_{5,8}$ and $DSM^{do}_{6,9}$, like the table below.

Variable	Value [MWh]
DSMup[4]	10.0
DSMup[5]	10.0
DSMup[6]	10.0

Variable	Value [MWh]
DSMdo[4,7]	10.0
DSMdo[5,8]	10.0
DSMdo[6,9]	10.0

One of the equations (2.8), (2.9) and (2.10) can be ignored, depending on the values of C^{up} and C^{do}.

- If $C^{up} > C^{do}$, then equation (2.10) covers for equation (2.8) and it can be ignored.
- If $C^{up} < C^{do}$, then equation (2.10) covers for equation (2.9) and it can be ignored.
- If $C^{up} = C^{do}$, then equations (2.8) and (2.9) cover for equation (2.10) and it can be ignored.

It can be also explained with a mathematical expression with assuming $C^{up} > C^{do}$:

$$DSM_{tt}^{up} + \sum_{t=tt-L}^{tt+L} DSM_{t,tt}^{do} \leq \max\{C^{up}, C^{do}\}$$

$$DSM_{tt}^{up} + \sum_{t=tt-L}^{tt+L} DSM_{t,tt}^{do} \leq C^{up}$$

$$DSM_{tt}^{up} \leq C^{up} - \sum_{t=tt-L}^{tt+L} DSM_{t,tt}^{do}$$

In the end, DSM_t^{up} needs to be less than C^{up} minus a value of $DSM_{t,tt}^{do}$s. It gets tighter than the equation (2.8), which could be ignored.

Chapter 3

urbs Model

urbs [7] is a linear optimization model for energy systems, written in python with a python library called "Coopr/Pyomo" mainly by Johannes Dorfner at Institute for Renewable and Sustainable Energy Systems, Technical University of Munich. It takes an excel worksheet as an input file and finds the minimum cost of energy system with satisfying the given demand. In the input file there are several types of data; such as the demand for given sites, the transmission details of the sites, buy-sell prices of the commodities, storage capacities, etc. The input file can be changed by user in order to test for different regions or countries. The model itself can be found in github and it is an open-source program, therefore it can be changed or used by anyone for any power-system in the world. After calculating the solution, urbs gives as output an other excel file for the sake of showing the results. Moreover to the excel worksheet, urbs also provides output pictures as graphics and comparison charts.

One of the charts of the ubs can be seen below and also a comparison chart is shown beneficial for understanding the differences between scenarios.

So far urbs is able to calculate and find a solution for a minimum cost of energy system with given input file as Bavarian Model, however a DSM capability is needed to be implement into urbs, meanwhile with adding the values of DSM constants into the input file. It is easy to implement DSM capabilities into the code, but urbs is a complex program and needed to deal with caution.

3.1 The Code pre-Implementation

Before implementing anything into urbs, the formulation which will be used in urbs, has to be checked, whether it is working properly or not. Therefore a discrete python program is coded and named testdsm.py. testdsm.py is just a program to see if the Zerrahn-Schill [3] formulation is working just as explained on the paper "On the representation of demand-side management in power system models". After 200 lines of code and coding a plot function, testdsm.py is able to run just as on the paper described. Some output pictures of this program can be found below.

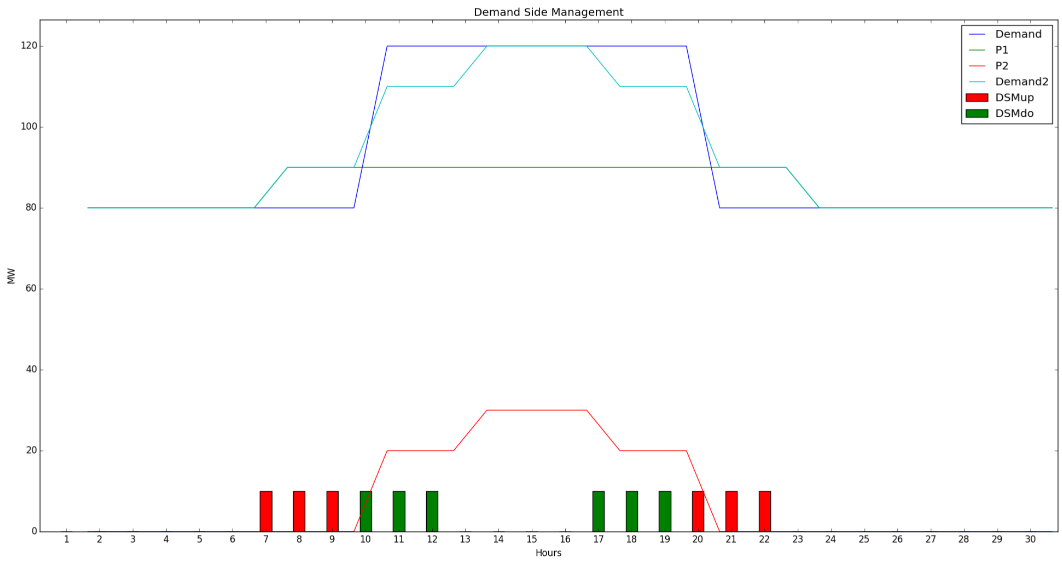


Figure 3.1 Power Generation & Demand Graph ($L = 3, \eta = 1, R = 1$)

In the figure 3.1 two different power generators, P1 & P2, are realized and also the given demand is shown with the dark blue line. The code tries to satisfy the demand with these two generators, and they both have a threshold capacity value of 90 MW. Only difference between both power generators is, P1 is much more cheaper than P2. The code calculates the optimum solution with generating the least power as possible from the generator P2 in order to get a minimum cost to satisfying the demand. At this point DSM capabilities starts working and they shift the high demand, between the hours 10 to 19. As it is also shown in the figure 3.1 DSM^{up} variables are increasing the demand before the peak time ($t \leq 10$), and also after the peak time ($t \geq 19$); so that DSM^{do} variables are able to decrease the demand between the peak times ($10 \leq t \leq 19$). In the end, the new demand line (demand2 in figure 3.1) is created.

The effects of the changing values of efficiency factor η can be found in the figure 3.2. It is important to be observe, that the DSM^{do} variables are exactly half of the DSM^{up} variables, while $\eta = 0.5$. In the second graph DSM^{up} variables are 0.8 times bigger than DSM^{do} variables, while $\eta = 0.8$. In other words efficiency factor is only there for setting up the values of DSM^{do} variables like in the equation (2.7) mentioned. After this results, it is clear, that the equations in Zerrahn-Schill [3] formulation function perfectly and they are ready to be implemented in urbs.

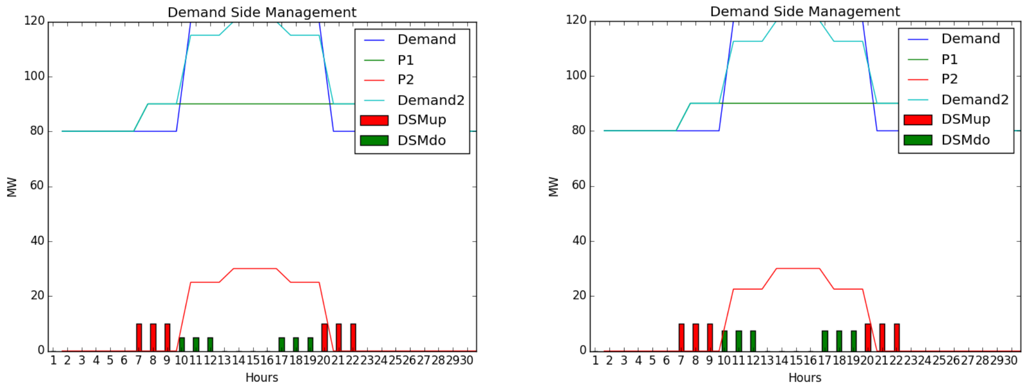


Figure 3.2 testdsm.py with $\eta = 0.5$ and $\eta = 0.8$

3.2 urbs with DSM

The implementation of DSM into *urbs Model* requires not only a python knowledge, but also proficiency with module "Coopr/Pyomo". Most aspects of DSM is needed to be coded into the urbs function `create_model(...)`. It is the function, where the power-system model is created in urbs. urbs code is modified and the following changes are made:

1. DSM constants, variables and constraints are added in the `create_model(...)` function.

DSM^{up} has two indexes, m.tm is for time index and m.sit is for site index. On the other hand DSM^{do} has three indexes. m.Tm is extra coded for DSM^{do}. It is the second time index just as described in the Zerrahn-Schill [3] formulation. The DSM constants take the values from the input worksheet. An extra excel sheet with the name "DSM" is created within the input file.

2. DSM constraints' functions are added into urbs.

In Pyomo environment the functions of constraints have to be individually coded, and in the constraints it is shown with `rule={name of the function}`. All of the five equations of Zerrahn-Schill [3] formulation are inside urbs as functions, which can be seen in the code below.

3. Finally the old demand constraint inside the urbs' function `res_vertex_rule(...)` is changed.

Old demand function worked with the principle, where the generation of power has to satisfy the demand; however now the generation of power has to satisfy the demand with the DSM capabilities, which are DSM_t^{up} and $DSM_{t,tt}^{do}$. It can be seen more detailed with following equations:

Old Demand Function: $PowerGen_t \geq Demand_t \quad \forall t$

New Demand Function: $$PowerGen_t \geq Demand_t + DSM_t^{up} - \sum_{t=tt-L}^{tt+L} DSM_{t,tt}^{do} \quad \forall t$$

3.2.1 Explanation of the implemented Code

Pyomo is a unique library, mainly focuses to optimizing problems. urbs contains many pyomo sets, variables and constraints. This section focuses explanation of the modified code within urbs, after implementation of DSM. The concrete model in urbs are coded with the letter `m`. Therefore; all of the pyomo components start with the letter `m`. Some of the modified code is shown below and rest can be found in Appendix.

`m.R` represents the recovery time R. With `data['dsm'].loc['R', 'Value']` python goes to the sheet "DSM" in the input worksheet and locates R. It sets the value of R to the number under the value column. Recovery time is designed to have integer values, for that reason it is written inside an `int()` function.

```
m.R = int(data['dsm'].loc['R', 'Value'])
```

`m.DSMup` is designed for representing the DSM_t^{up} variable. It is mentioned in the formulation, that DSM_t^{up} has only one index. However; urbs calculates different solutions for different sites and DSM variables should also work for several sites. Therefore another index, called `m.sit` is added to the variable DSM_t^{up}. `m.tm` and `m.sit` are the indexes of `m.DSMup`. The starting values of `m.DSMup` are equal to 0 and they are defined with `Initialize=0` . `within=pyomo.NonNegativeReals` assures, that `m.DSMup` can only have positive values.

```
m.DSMup = pyomo.Var(
    m.tm, m.sit,
    initialize=0,
    within=pyomo.NonNegativeReals,
    doc='DSM Decreaser')
```

Constraints are the equations, which set the rules for the variables. For instance, the equation (2.8) of Zerrahn-Schill [3] formulation assures, that the DSM_t^{up} variables have to be less than a threshold value of C^{up}. Here it is represented with `m.Cup`. Most important thing to know is, that pyomo constraints have to have a function to be able to work. It is defined with `rule={name of the function}`.

Moreover, the function `dsmup_constraint_rule(m, tm, sit)` takes three input parameters. `m` is the model itself, `m.tm` is the time index and `m.sit` is the site index of the model. While the solver works, it calls the constraint function and sets the values according to the return inequality. The return inequality should give true as output every time. The solver also calls the functions several times. For instance, if `m.tm` is defined from 1 to 20 and there is three sites in the input file, then the solver calls this function 60 (20×3) times.

```
m.dsmupConstraint = pyomo.Constraint(
    m.tm, m.sit,
    rule=dsmup_constraint_rule,
    doc='Equation 2.2')

#Equation 2.2
def dsmup_constraint_rule(m, tm, sit):
    return m.DSMup[tm,sit] <= m.Cup
```

3.2.2 Runtime Comparison of urbs

In this section changes of the urbs' runtime due to different values of DSM constants are shown. First of all it is important to know, how urbs works. It is explained in following steps:

1) urbs code is compiled via python command prompt.

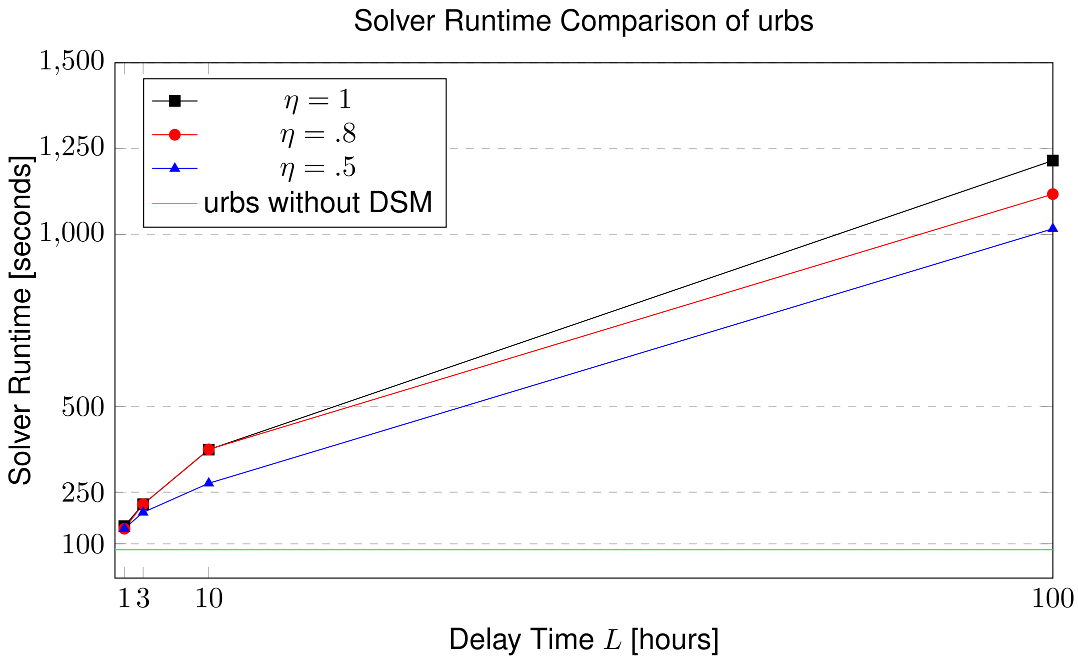
2) Data inside the input file is transferred to a dataframe.

3) GLPK LP/MIX Solver, v4.56 starts to calculate the optimum solution for urbs with given input data.

4) After calculating the solution, output worksheet is created by urbs.

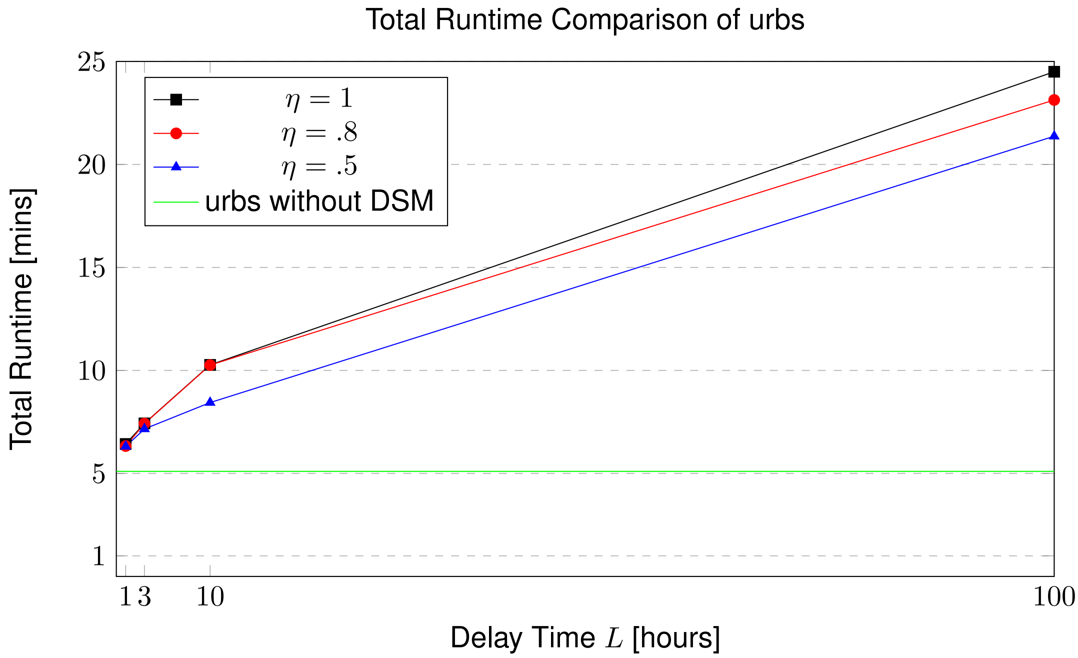
5) Finally, plot function of urbs is called and the output charts are created.

Solver runtime comparison of urbs refers to the spent time, which has passed, while solver calculates a solution for given input data. Below it is shown that the delay time L has a great effect to the solver runtime, on the other hand the effect of the efficiency factor η to the runtime is much more smaller. However, there is a hidden knowledge between these two DSM constants, which is that when the delay time is getting longer, the effect of the efficiency factor to the runtime increases. Also in the graph, it is easy to see that the gap between the lines is increasing with longer delay times.



Longer delay times cause more variations of load shifting and the GLPK solver should search and find the optimum between these variations, that is the main reason of the increased solver runtime due to increasing values of L. Efficiency factor should not have a big consequence on solver runtime, however it does make a difference between 4.5% - 16.33%.

In the next graph total runtime comparison of urbs is shown. Total runtime means the passed time from the beginning of step **1** until the end of step **5**, which are mentioned above. Delay time L and efficiency factor η have the same effect just as they had on solver runtime. It is important to mention, just implementing two variables into urbs have an effect of increased runtime about 20mins.



Memory usage is the required memory, which is accessed by solver, while calculating a solution. Starting from 91.5 Mbs (without DSM), with longer delay times it can be increased to 363.2 Mbs. Although it was not **hard for the computer**, which is used for this test, it can be problematic when urbs is runned for a year or longer with more data. Furthermore, decreasing values of efficiency factor have no impact of memory usage as expected.

Delay Time L [hours]	Efficiency Factor η	Memory Usage [Mbs]
—	—	91.5
1	1 / 0.8 / 0.5	101.2
3	1 / 0.8 / 0.5	107.7
10	1 / 0.8 / 0.5	130.7
100	1 / 0.8 / 0.5	363.2

All tests are performed with input file "Bavarian Model.xlsx" on Asus UX32V NoteBook, Intel® Core^{TM} i7-3517U @1.90GHz 2.40GHz from the scenario "Base" in time periods "aut", a.k.a. "Autumn" .

Chapter 4

Results

After several runs of the model urbs, output data is provided and used to create the figure 4.1. It shows percentage of the total cost differences between urbs with and without DSM capabilities. Of course when DSM capabilities are used, the total cost of the power-system model reduces.

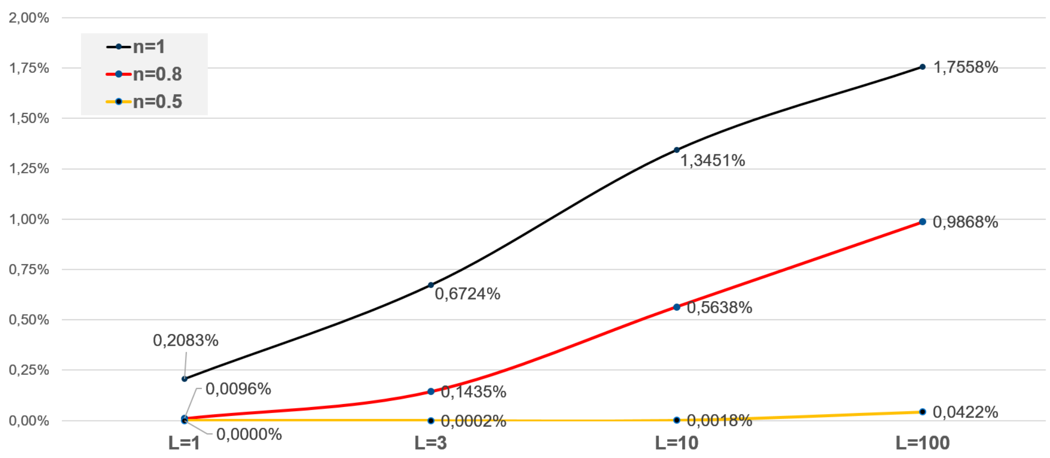


Figure 4.1 Profit between urbs with & without DSM

In order to get acquired results, urbs is ran with different values of the DSM constants. It is obvious, that when the delay time L increases, DSM variables may shift the load more and as a result profit gain increases. With the same principle, the efficiency factor η has also same effect on the results.

urbs uses different energy sources to satisfy the demand; such as biomass, gas, hydro, wind, coal, and gud plants. When DSM capabilities are installed, there are some differences. Following figures 4.2 and 4.3 are shown, that the usage of energy sources, when urbs is run with and without DSM capabilities.

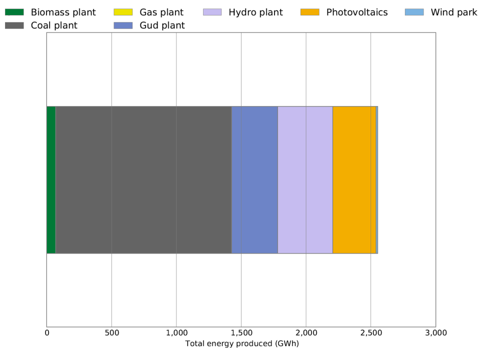


Figure 4.2 urbs Comparison Chart without DSM

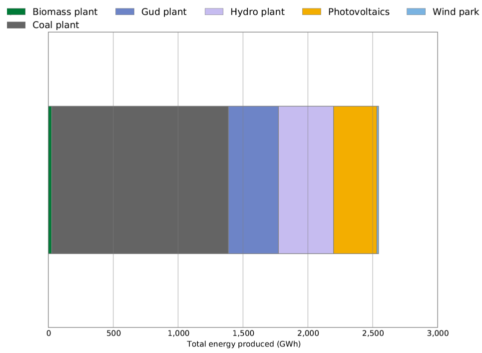


Figure 4.3 urbs Comparison Chart with DSM

There is a loss on the generation of energy due to implementation of DSM capabilities into urbs. It is hardly to be seen in the figures 4.2 and 4.3. However with the calculated numbers the difference is 0.33%. The usage of the gas plant is not necessary, when urbs runs with DSM. The gap of the gas plant, and the loss from biomass plant is compensated by gud and coal plants, like it is shown in figure 4.3.

It is mentioned before, that DSM can help integration of renewable energy sources. In the results DSM does not clearly help integrating them, but it is because urbs model mainly focuses on minimizing total cost of the energy system. However, the loss on the generation of energy shows that for the same demand, it is not necessary to generate that much power.

More Detailed Results with L=1, 3, 10, 100

The outcome can show variability depending on the given time steps, while urbs is run. For better understanding, urbs is run for a quarter of a year with following DSM parameters:

- Scenerio Base $L = 1$ h; $\eta = 1$; $R = 1$ h; C^{do} & $C^{up} = 150$ MWh
- Scenerio L3 $L = 3$ h; $\eta = 1$; $R = 1$ h; C^{do} & $C^{up} = 150$ MWh
- Scenerio L10 $L = 10$ h; $\eta = 1$; $R = 1$ h; C^{do} & $C^{up} = 150$ MWh
- Scenerio L100 $L = 100$ h; $\eta = 1$; $R = 1$ h; C^{do} & $C^{up} = 150$ MWh

The scenarios L3, L10 and L100 are compared with the scenario Base, which has the minimum value of the delay time $L = 1$. In the figure 4.4 it is shown, that total costs are reducing with increasing values of the delay time L. There are also differences on the produced energy levels. Just as the total cost comparison, produced energy is decreasing with the increasing values of the delay time for same energy demand. It is easier to see the differences between scenarios in the tables 4.1 and 4.2. During the calculation of profit, following equation is used:

$$Profit_{l100/l10/l3} = Total_{base} - Total_{l100/l10/l3}$$

The positive values of the profit tab can be interpreted as either spending less money or producing less energy for the same demand depending on the analyzed table.

	Fix	Fuel	Inv	Purchase	Revenue
base	1,910499841	2,916002177	0,789190267	0	0
l100	1,896529554	2,88183881	0,757726481	0	0
l10	1,902593053	2,888525533	0,772965515	0	0
l3	1,90725621	2,906784977	0,78132585	0	0
	Var	**Total**	**Profit[1e9 €/a]**	**Profit[%]**	
base	0,268534502	5,615692285	0	0,00%	
l100	0,266710039	5,536094845	0,07959744	1,42%	
l10	0,267169155	5,564084101	0,051608185	0,92%	
l3	0,267869684	5,595367037	0,020325249	0,36%	

Table 4.1 Total Cost Comparison with $L = 1, 3, 10, 100$ Hours

	Biomass plant	Coal plant	Gas plant	Gud plant	Hydro plant
base	578,2722005	16770,57847	92,13649276	3402,769889	2010,870908
l100	389,9475254	16892,00232	26,54828244	3498,013814	2010,870908
l10	467,7706081	16992,23191	31,96260691	3339,238328	2010,870908
l3	511,8959702	16798,04576	75,4862791	3453,271371	2010,870908
	Photovoltaics	**Wind park**	**Total**	**Profit[GWh]**	**Profit[%]**
base	1098,133183	429,3071498	22854,62796	0	0,00%
l100	1098,133183	429,3071498	22817,38285	37,24510579	0,16%
l10	1098,133183	429,3071498	22842,07436	12,55359544	0,05%
l3	1098,133183	429,3071498	22849,57028	5,057671977	0,02%

Table 4.2 Energy Sums Comparison with $L = 1, 3, 10, 100$ Hours

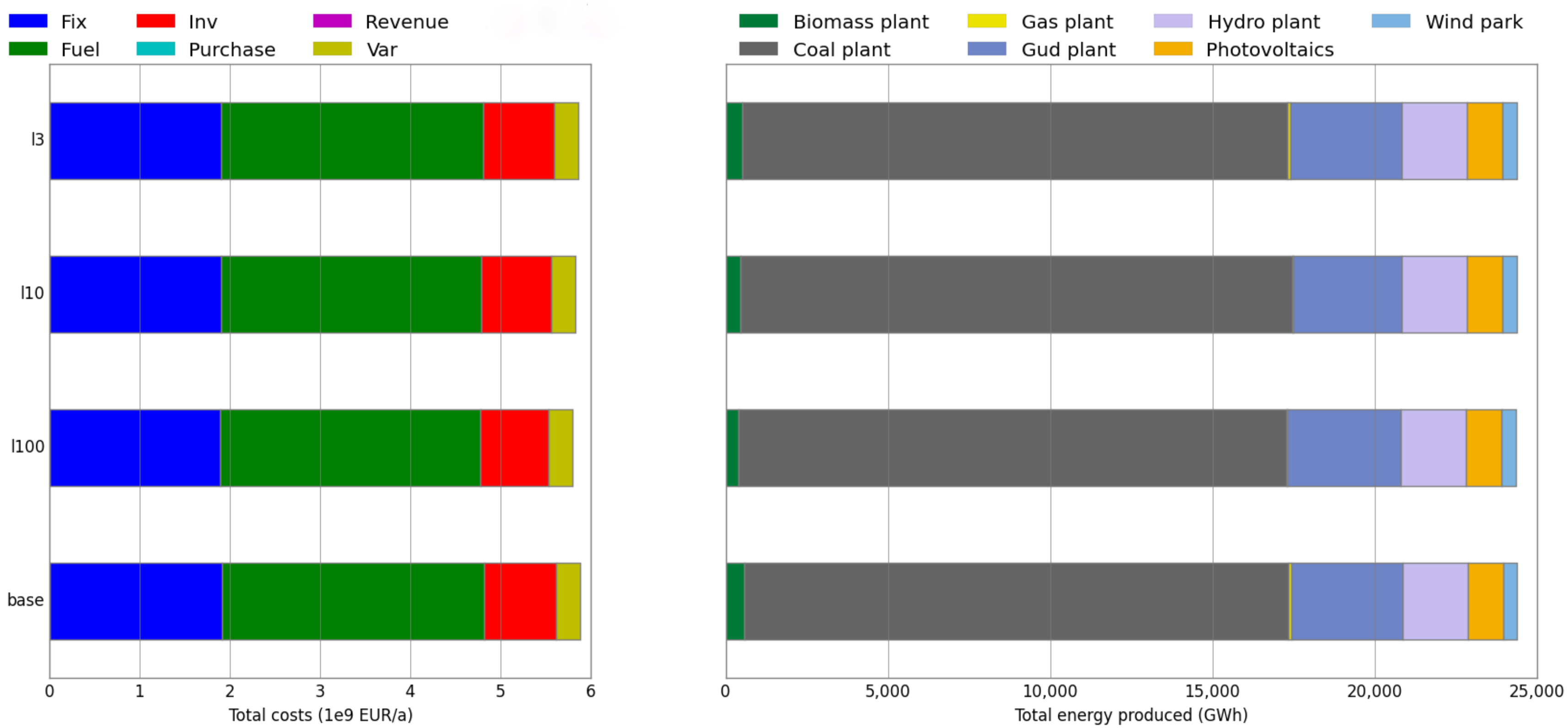


Figure 4.4 Comparison with $L = 1, 3, 10, 100$ Hours

Chapter 5

Conclusion

This thesis mainly focuses showing the differences between with and without DSM usage on a power-system model. As a power-system model, urbs from the Institute for Renewable and Sustainable Energy Systems, Technical University of Munich is used. It is expected with the usage of DSM, total cost for satisfying the demand within the model should reduce.

Implementation of the Zerrahn-Schill's DSM formulation in python language is firstly needed to achieve this goal. After several versions of DSM code, bugs are fixed and urbs is modified with that code. The variables and parameters of the formulation is rearranged to be able fit in urbs syntax. Finally the resulting errors are corrected and the outcome results are shown.

In the section 2 of the thesis, the inside knowledge of the DSM capabilities is explained. The mathematical documentation of load shifting, the differences between formulation Göransson et al. and Zerrahn-Schill are discussed.

The model urbs is modified in order to represent DSM aspects within it's calculations. The pre steps of implementing DSM are clarified. The runtime comparison of the model urbs with and without DSM is also revealed in the section 3. Finally the results are evaluated in section 4.

Altogether, it can be said, that after implementing DSM to the model urbs, even though runtime of urbs increases with the usage of DSM, there is a clear improvement on levels of profit in contrast of runtime. In addition, memory usage of urbs is increased with the usage of DSM. It should be noted, that longer time periods can cause a memory error in the machine, which urbs runs. Upcoming researches can analyze more to the memory problems of urbs and maybe can find a solution for this dilemma. Without being sure, the library Coopr/Pyomo is suspected for being the main reason of these memory problems.

Appendix

Sets, Indexes, Parameters and Variables of Formulations

Item	Description	Unit
Sets and indexes		
$t, tt \in T$	Time periods	Hours
$l \in L$	Delay time	Hours
Parameters of the Göransson et al. formulation		
C^{dd}	Installed capacity for hourly demand delayed	MWh
C^{ds}	Installed capacity for hourly demand served	MWh
Variables of the Göransson et al. formulation		
dd_t	Hourly demand delayed	MWh
dh_t	Cumulative hourly demand put on hold	MWh
ds_t	Hourly demand served	MWh
Parameters of the Zerrahn-Schill formulation		
C^{do}	Installed capacity for hourly downward shifts	MWh
C^{up}	Installed capacity for hourly upward shifts	MWh
η	Efficiency factor	—
R	Recovery time	Hours
Variables of the Zerrahn-Schill formulation		
$DSM^{do}_{t,tt}$	Hourly downward load shifts for hour t in hour tt	MWh
DSM^{up}_t	Hourly upward load shifts	MWh

The Modified Code; Additional Functionality to the urbs Package

```
def create_model(data, timesteps=None, dt=1):
...
    #Demand Side Management Modelled Time Steps
    m.Tm = pyomo.Set(
        within=m.t,
        initialize=m.timesteps[1:],
        ordered=True)
    #Demand Side Management Constants
    m.L = int(data['dsm'].loc['L', 'Value'])
    m.n = float(data['dsm'].loc['n', 'Value'])
    m.R = int(data['dsm'].loc['R', 'Value'])
    m.Cdo = float(data['dsm'].loc['Cdo', 'Value'])
    m.Cup = float(data['dsm'].loc['Cup', 'Value'])
    #Demand Side Management Variables
    m.DSMup = pyomo.Var(
        m.tm, m.sit,
        initialize=0,
        within=pyomo.NonNegativeReals,
        doc='DSM Decreaser')
    m.DSMdo = pyomo.Var(
        m.tm, m.Tm, m.sit,
        initialize=0,
        within=pyomo.NonNegativeReals,
        doc='DSM Increaser')
    #Demand Side Management Constraints
    m.dsmupdoConstraint = pyomo.Constraint(
        m.tm, m.sit,
        rule=dsmupdo_constraint_rule,
        doc='Equation 2.1')
    m.dsmupConstraint = pyomo.Constraint(
        m.tm, m.sit,
        rule=dsmup_constraint_rule,
        doc='Equation 2.2')
    m.dsmdoConstraint = pyomo.Constraint(
        m.Tm, m.sit,
        rule=dsmdo_constraint_rule,
        doc='Equation 2.3')
    m.C2Constraint = pyomo.Constraint(
        m.Tm, m.sit,
        rule=C2_constraint_rule,
        doc='Equation 2.4')
    m.dsmup2Constraint = pyomo.Constraint(
        m.tm, m.sit,
        rule=dsmup2_constraint_rule,
        doc='Equation 2.5')
...
```

```
#Demand Side Management Functions
#Equation 2.1
def dsmupdo_constraint_rule(m, tm, sit):
    if tm <= m.timesteps[0] + m.L:
        return sum(m.DSMdo[tm,T,sit] for T in range(m.timesteps[0] + 1, tm+1+m.L)) \
        == m.DSMup[tm,sit] * m.n
    elif tm >= m.timesteps[0]
    1 + m.L and tm <= m.timesteps[-1] - m.L:
        return sum(m.DSMdo[tm,T,sit] for T in range(tm-m.L, tm+1+m.L)) \
        == m.DSMup[tm,sit] * m.n
    else:
        return sum(m.DSMdo[tm,T,sit] for T in range(tm-m.L, m.timesteps[-1] + 1)) \
        == m.DSMup[tm,sit] * m.n
#Equation 2.2
def dsmup_constraint_rule(m, tm, sit):
    return m.DSMup[tm,sit] <= m.Cup
#Equation 2.3
def dsmdo_constraint_rule(m, Tm, sit):
    if Tm <= m.timesteps[0] + m.L:
        return sum(m.DSMdo[t,Tm,sit] for t in range(m.timesteps[0] + 1, Tm+1+m.L)) \
        <= m.Cdo
    elif Tm >= m.timesteps[0] + 1 + m.L and Tm <= m.timesteps[-1] - m.L:
        return sum(m.DSMdo[t,Tm,sit] for t in range(Tm-m.L, Tm+1+m.L)) \
        <= m.Cdo
    else:
        return sum(m.DSMdo[t,Tm,sit] for t in range(Tm-m.L, m.timesteps[-1] + 1)) \
        <= m.Cdo
#Equation 2.4
def C2_constraint_rule(m, Tm, sit):
    if Tm <= m.timesteps[0] + m.L:
        return max(m.Cup, m.Cdo) >= m.DSMup[Tm,sit] + \
        sum(m.DSMdo[t,Tm,sit] for t in range(m.timesteps[0] + 1, Tm+1+m.L))
    elif Tm >= m.timesteps[0] + 1 + m.L and Tm <= m.timesteps[-1] - m.L:
        return max(m.Cup, m.Cdo) >= m.DSMup[Tm,sit] + \
        sum(m.DSMdo[t,Tm,sit] for t in range(Tm-m.L, Tm+1+m.L))
    else:
        return max(m.Cup, m.Cdo) >= m.DSMup[Tm,sit] + \
        sum(m.DSMdo[t,Tm,sit] for t in range(Tm-m.L, m.timesteps[-1] + 1))
#Equation 2.5
def dsmup2_constraint_rule(m, tm, sit):
    if tm + m.R <= m.timesteps[-1] + 1:
        return sum(m.DSMup[tm,sit] for t in range(tm, tm+m.R)) \
        <= m.Cup * m.L
    else:
        return sum(m.DSMup[tm,sit] for t in range(tm, m.timesteps[-1] + 1)) \
        <= m.Cup * m.L
```

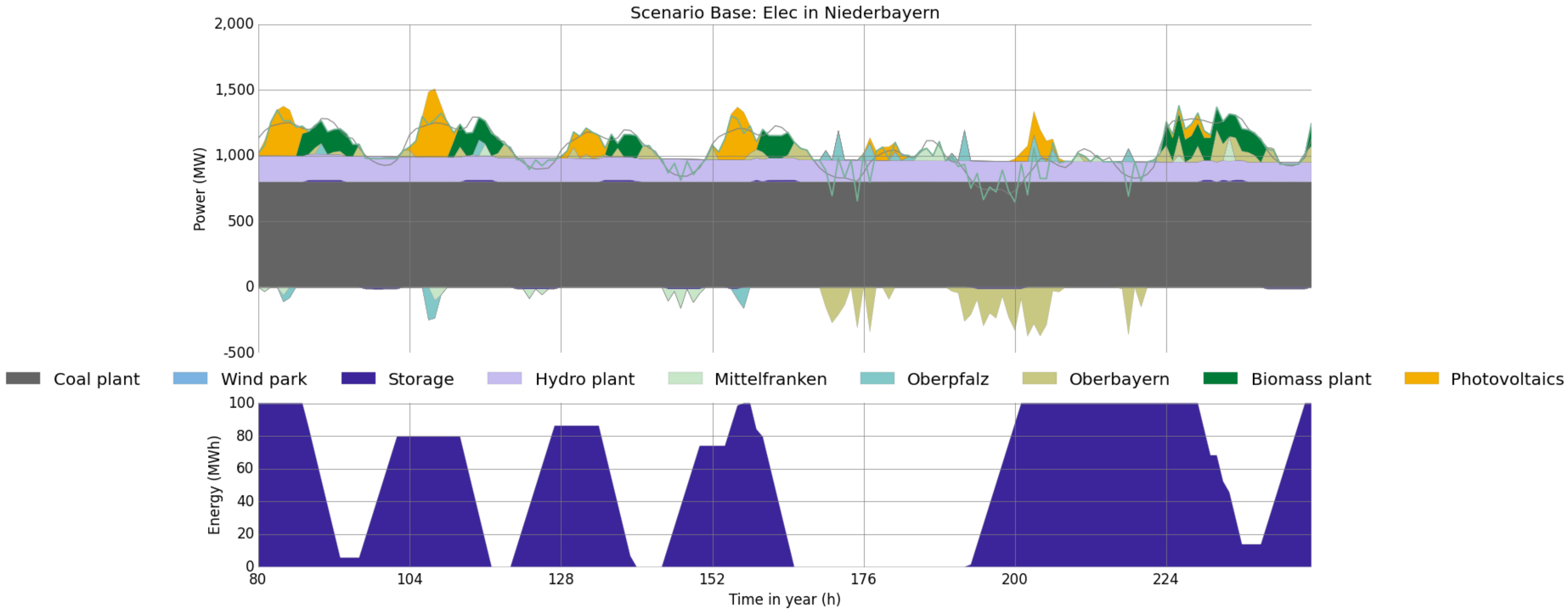
```
#Demand Function
def res_vertex_rule(m, tm, sit, com, com_type):
...
    #New Demand with Demand Side Management
    if com in m.com_demand:
        try:
            if tm <= m.timesteps[0] + m.L:
                power_surplus -= m.demand.loc[tm][sit, com] + m.DSMup[tm,sit] \
                - sum(m.DSMdo[T,tm,sit] for T in \
                range(m.timesteps[0] + 1, tm+m.L+1))
            elif tm >= m.timesteps[0] + 1 + m.L and tm <= m.timesteps[-1] - m.L:
                power_surplus -= m.demand.loc[tm][sit, com] + m.DSMup[tm,sit] \
                - sum(m.DSMdo[T,tm,sit] for T in \
                range(tm-m.L, tm+1+m.L))
            else:
                power_surplus -= m.demand.loc[tm][sit, com] + m.DSMup[tm,sit] \
                - sum(m.DSMdo[T,tm,sit] for T in \
                range(tm-m.L, m.timesteps[-1] + 1))
        except KeyError:
            pass
...
```

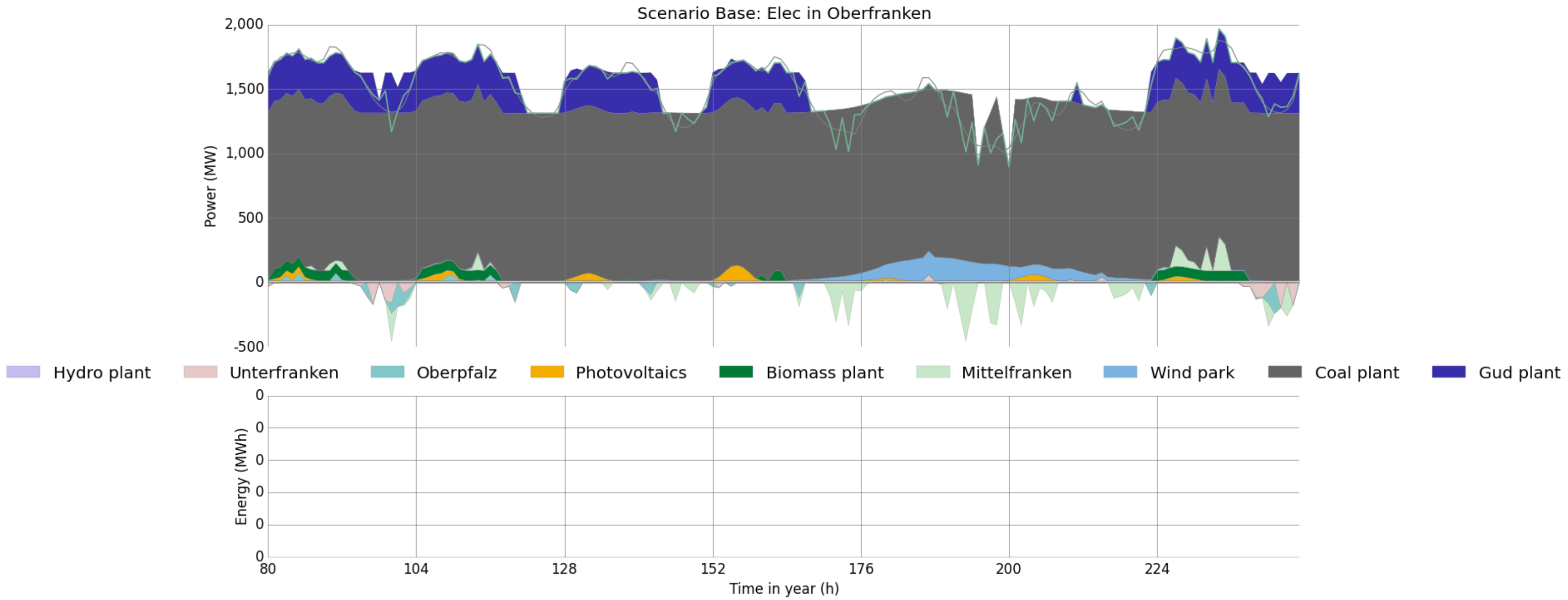
urbs' Output Images on Following Pages

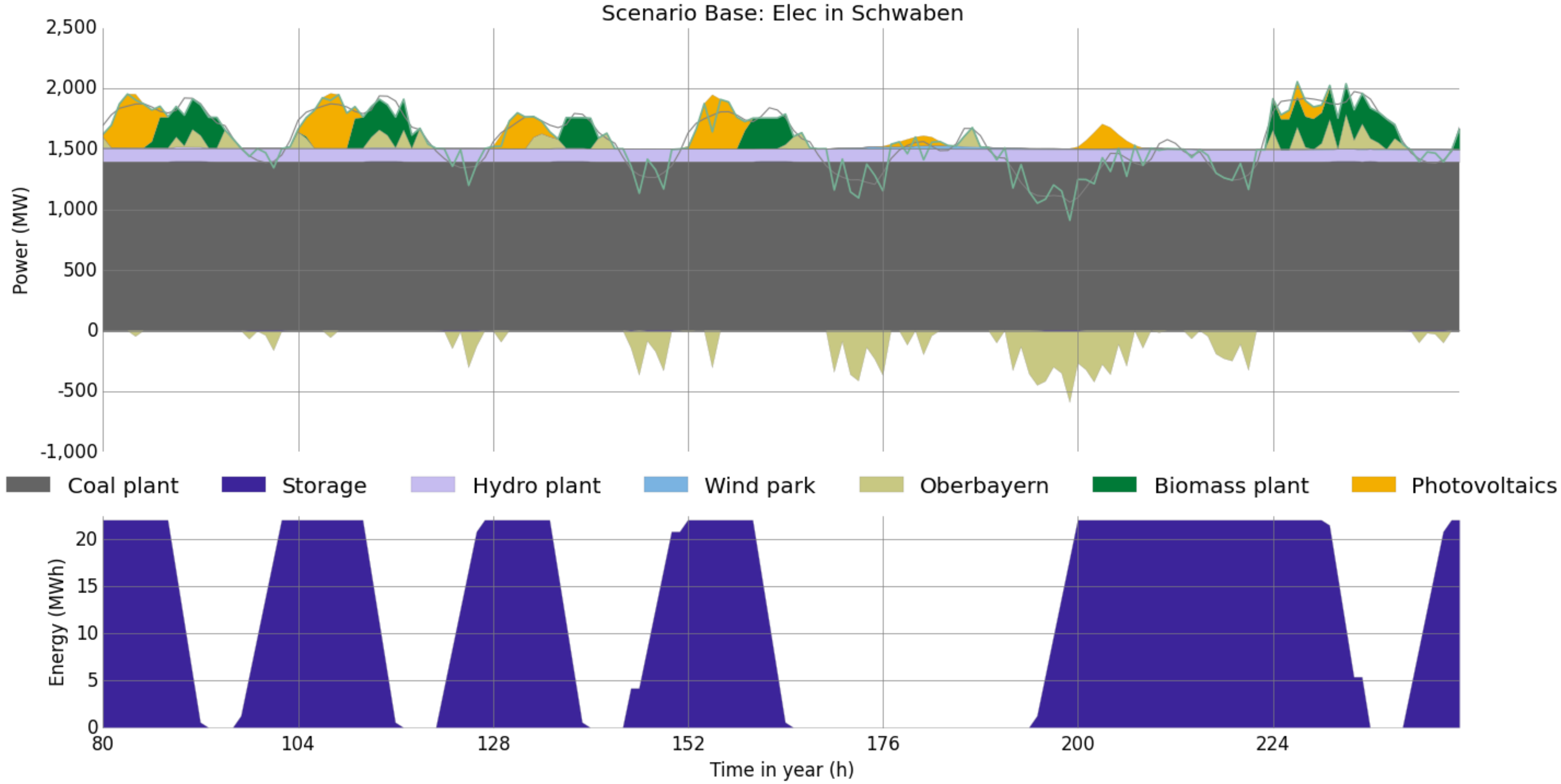
The given values of parameters:

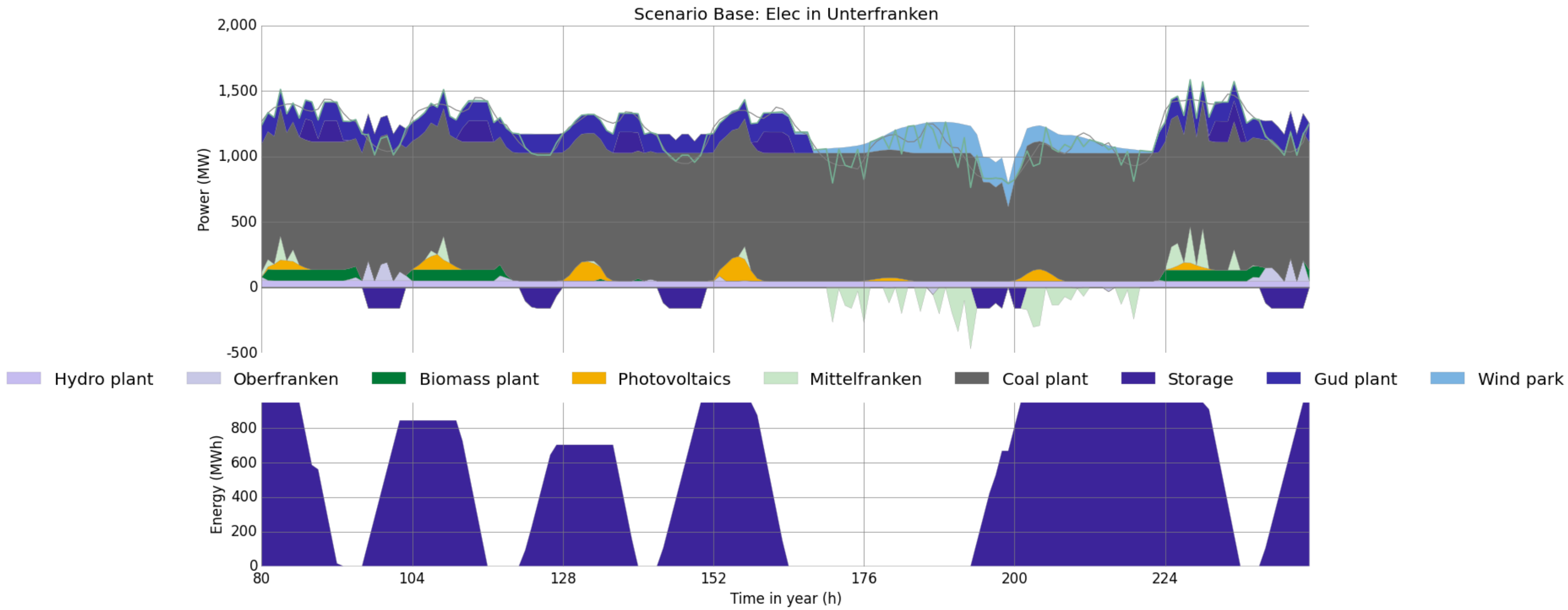
- Scenerio Base $L = 1$ h; $\eta = 1$; $R = 1$ h; C^{do} & $C^{up} = 150$ MWh
- Scenerio L3 $L = 3$ h; $\eta = 1$; $R = 1$ h; C^{do} & $C^{up} = 150$ MWh
- Scenerio L10 $L = 10$ h; $\eta = 1$; $R = 1$ h; C^{do} & $C^{up} = 150$ MWh
- Scenerio L100 $L = 100$ h; $\eta = 1$; $R = 1$ h; C^{do} & $C^{up} = 150$ MWh

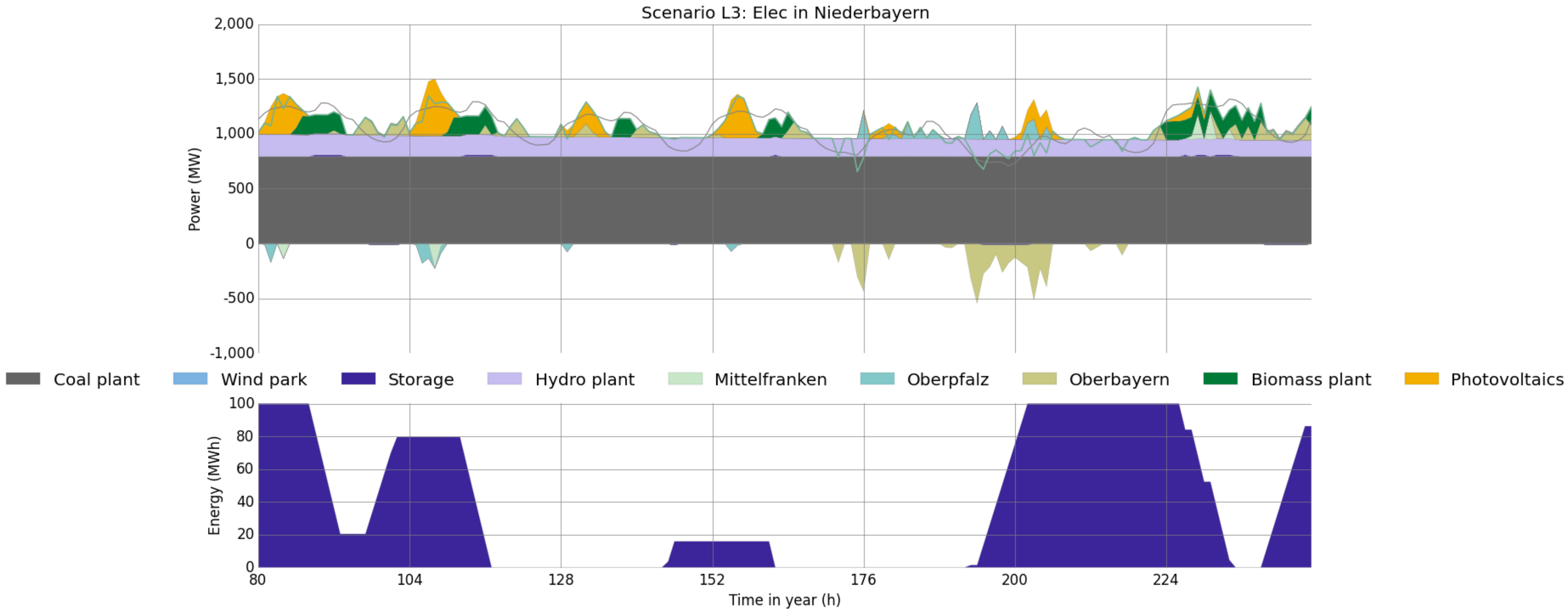
Not all of the output images are included. The missing ones can be found in the additional DVD (under /thesis-latex/img/urbs).


Scenario Base: Elec in Niederbayern
Power (MW)
2,000
1,500
1,000
500
0
-500
Coal plant
Wind park
Storage
Hydro plant
Mittelfranken
Oberpfalz
Oberbayern
Biomass plant
Photovoltaics
Energy (MWh)
100
80
60
40
20
0
80
104
128
152
176
200
224
Time in year (h)

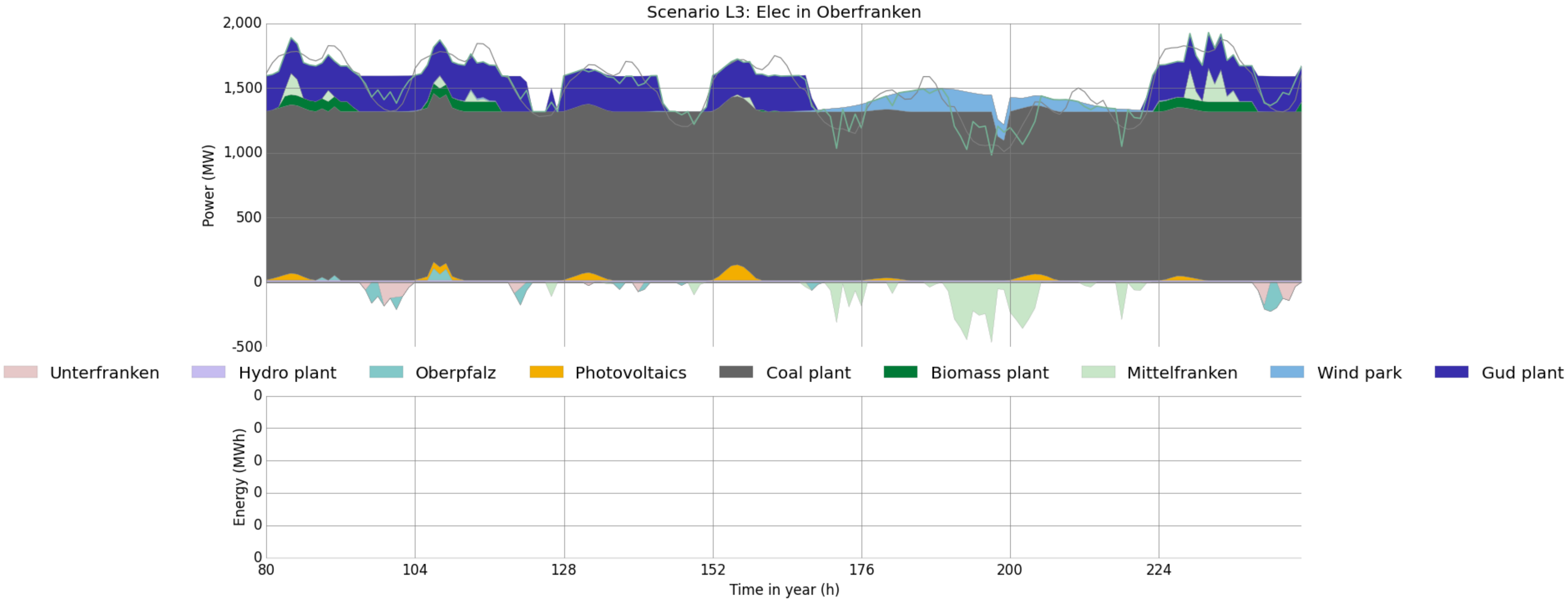

Scenario Base: Elec in Oberfranken
2,000
1,500
1,000
500
0
-500
Power (MW)
Hydro plant
Unterfranken
Oberpfalz
Photovoltaics
Biomass plant
Mittelfranken
Wind park
Coal plant
Gud plant
Energy (MWh)
0
0
0
0
0
0
80
104
128
152
176
200
224
Time in year (h)

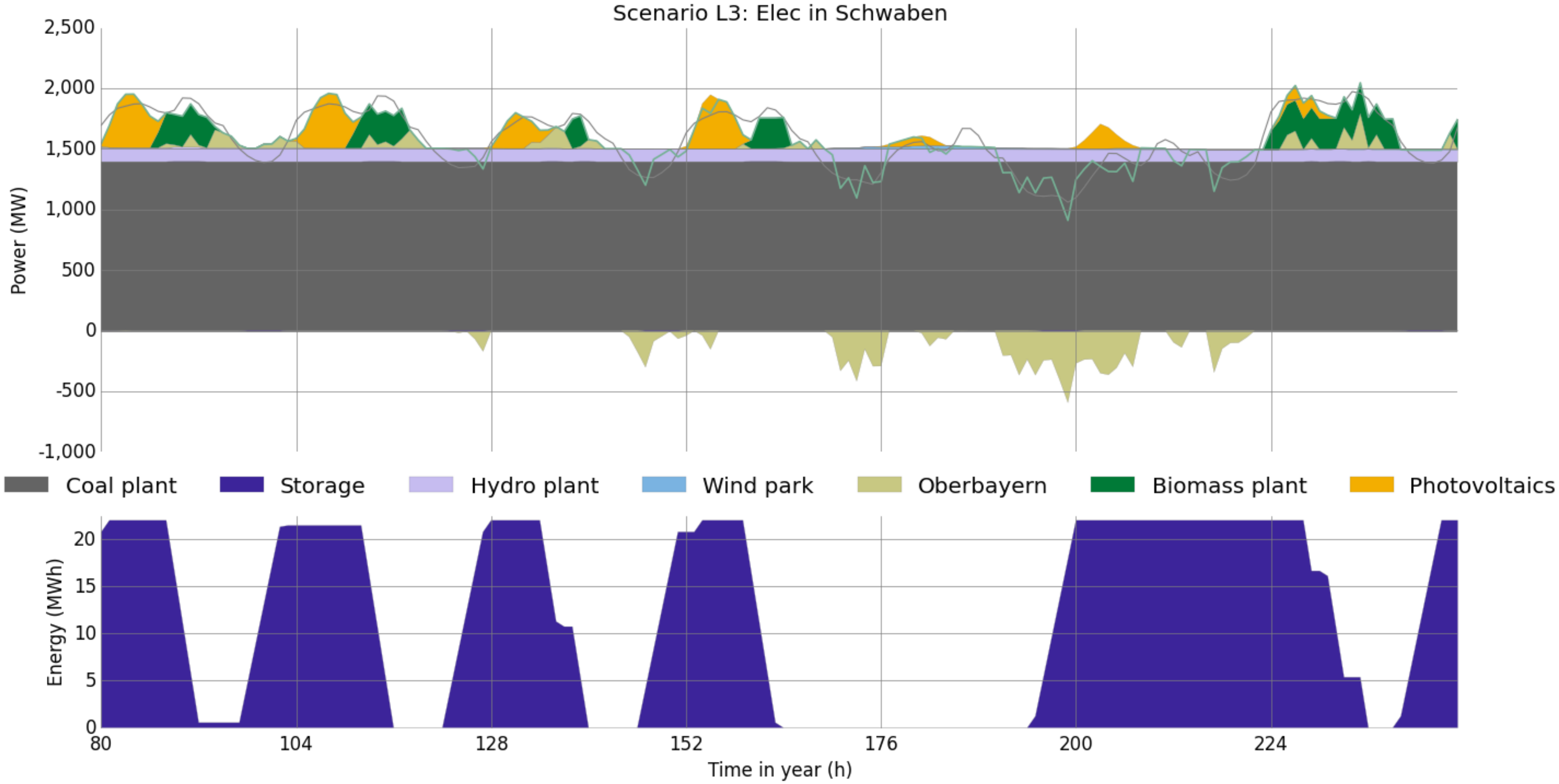

Scenario Base: Elec in Schwaben
Power (MW)
2,500
2,000
1,500
1,000
500
0
-500
-1,000
Coal plant
Storage
Hydro plant
Wind park
Oberbayern
Biomass plant
Photovoltaics
Energy (MWh)
20
15
10
5
0
80
104
128
152
176
200
224
Time in year (h)

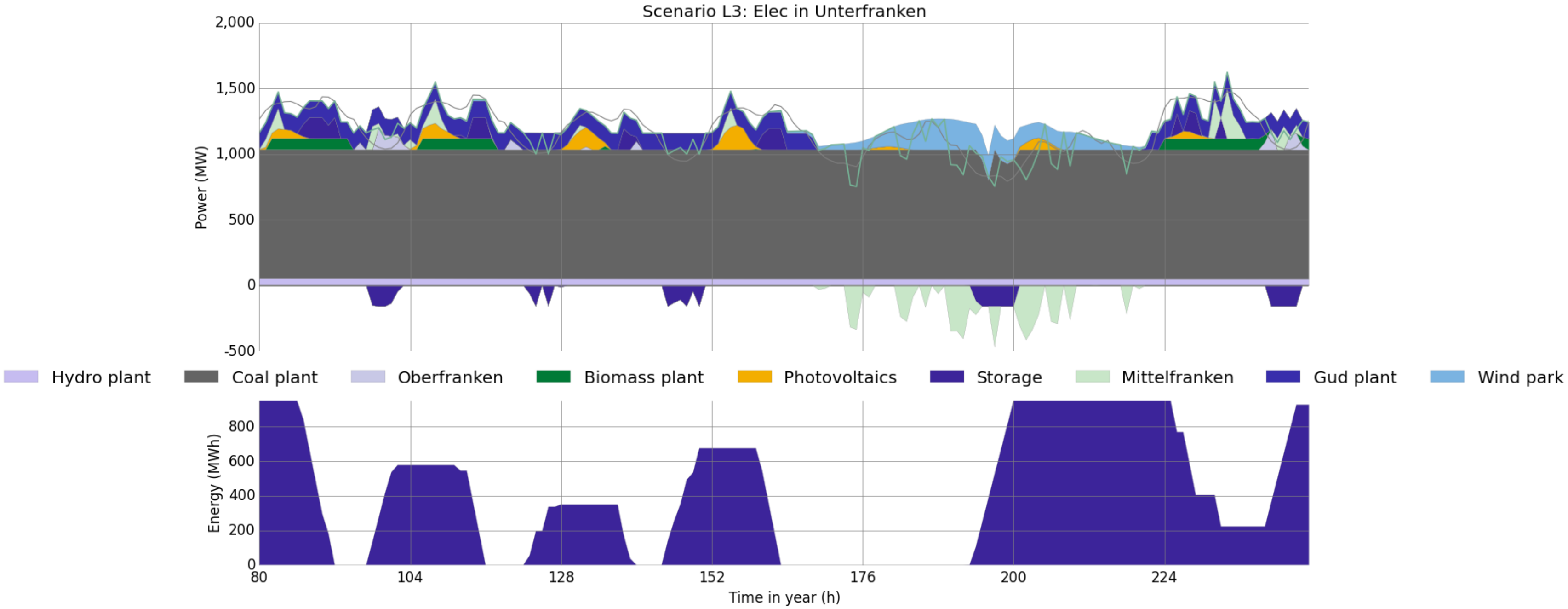

Scenario Base: Elec in Unterfranken
Power (MW)
2,000
1,500
1,000
500
0
-500
Hydro plant
Oberfranken
Biomass plant
Photovoltaics
Mittelfranken
Coal plant
Storage
Gud plant
Wind park
Energy (MWh)
800
600
400
200
0
80
104
128
152
176
200
224
Time in year (h)

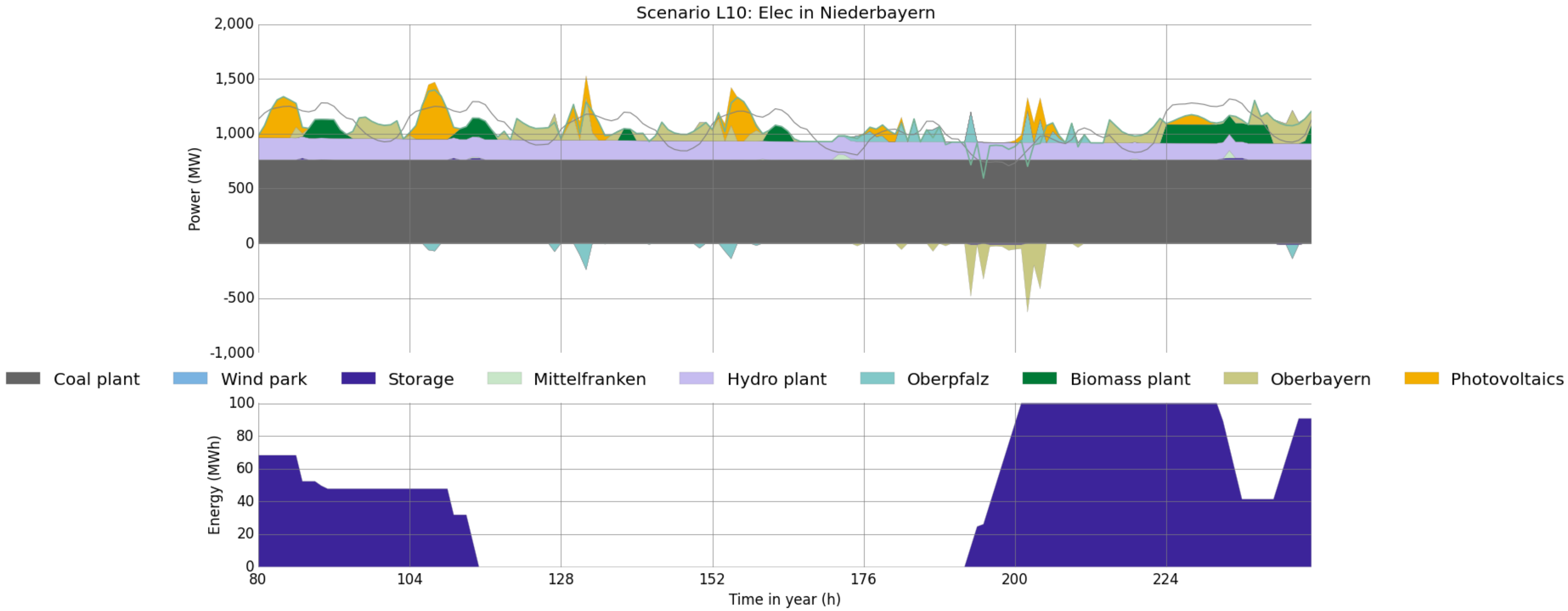


Scenario L3: Elec in Niederbayern
Power (MW)
2,000
1,500
1,000
500
0
-500
-1,000
Coal plant
Wind park
Storage
Hydro plant
Mittelfranken
Oberpfalz
Oberbayern
Biomass plant
Photovoltaics
Energy (MWh)
100
80
60
40
20
0
80
104
128
152
176
200
224
Time in year (h)

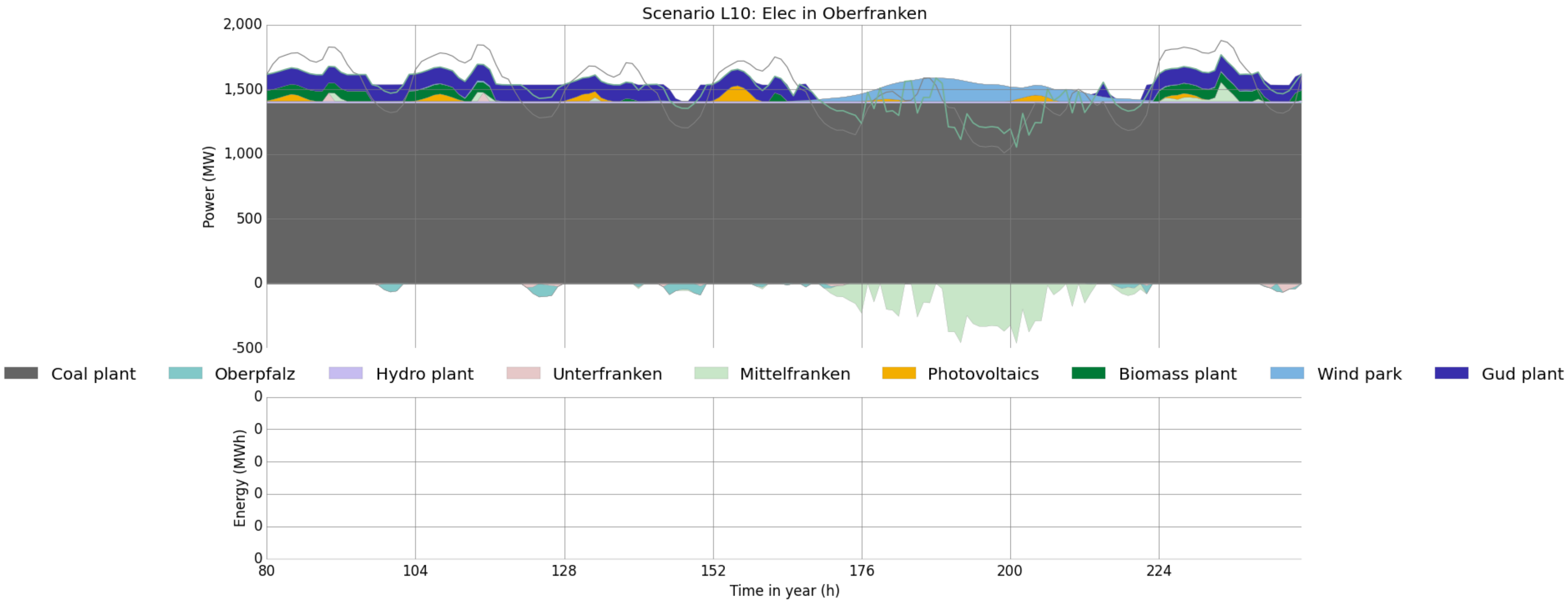

Scenario L3: Elec in Oberfranken
Power (MW)
2,000
1,500
1,000
500
0
-500
Unterfranken
Hydro plant
Oberpfalz
Photovoltaics
Coal plant
Biomass plant
Mittelfranken
Wind park
Gud plant
Energy (MWh)
0
80
104
128
152
176
200
224
Time in year (h)


Scenario L3: Elec in Schwaben
Power (MW)
2,500
2,000
1,500
1,000
500
0
-500
-1,000
Coal plant
Storage
Hydro plant
Wind park
Oberbayern
Biomass plant
Photovoltaics
Energy (MWh)
20
15
10
5
0
80
104
128
152
176
200
224
Time in year (h)

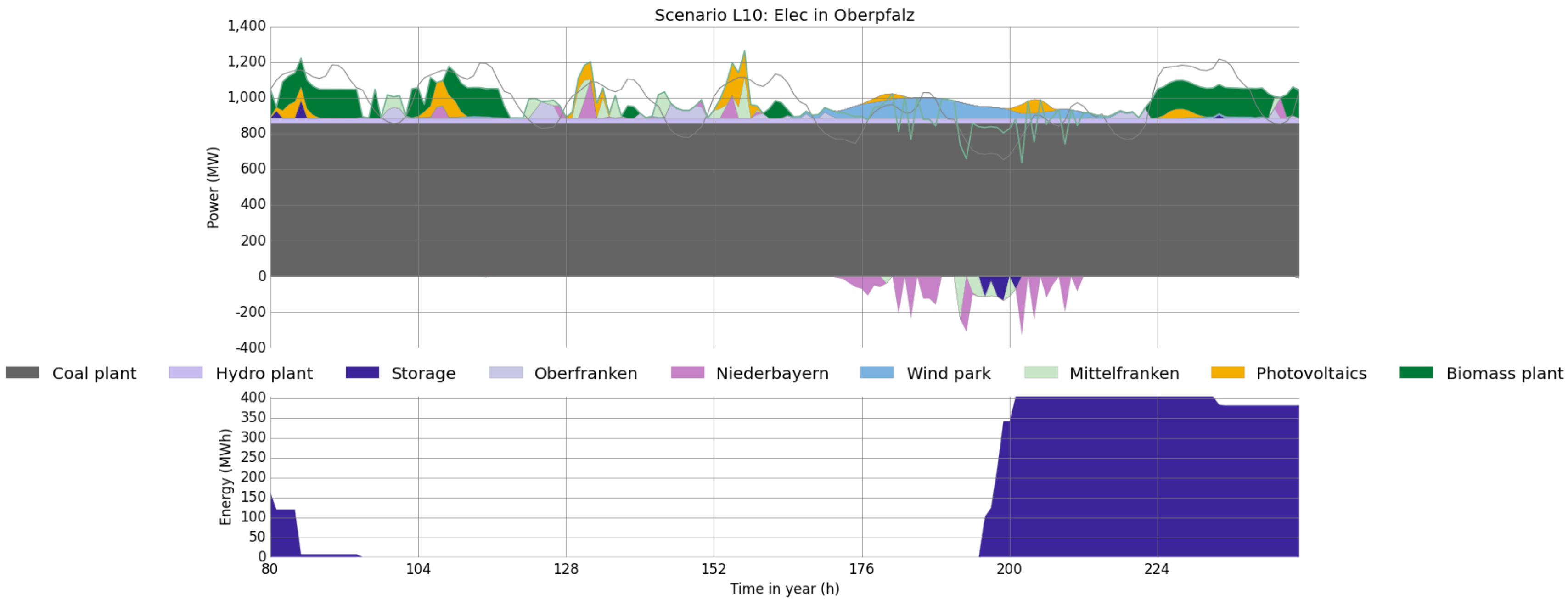

Scenario L3: Elec in Unterfranken
Power (MW)
2,000
1,500
1,000
500
0
-500
Hydro plant
Coal plant
Oberfranken
Biomass plant
Photovoltaics
Storage
Mittelfranken
Gud plant
Wind park
Energy (MWh)
800
600
400
200
0
80
104
128
152
176
200
224
Time in year (h)

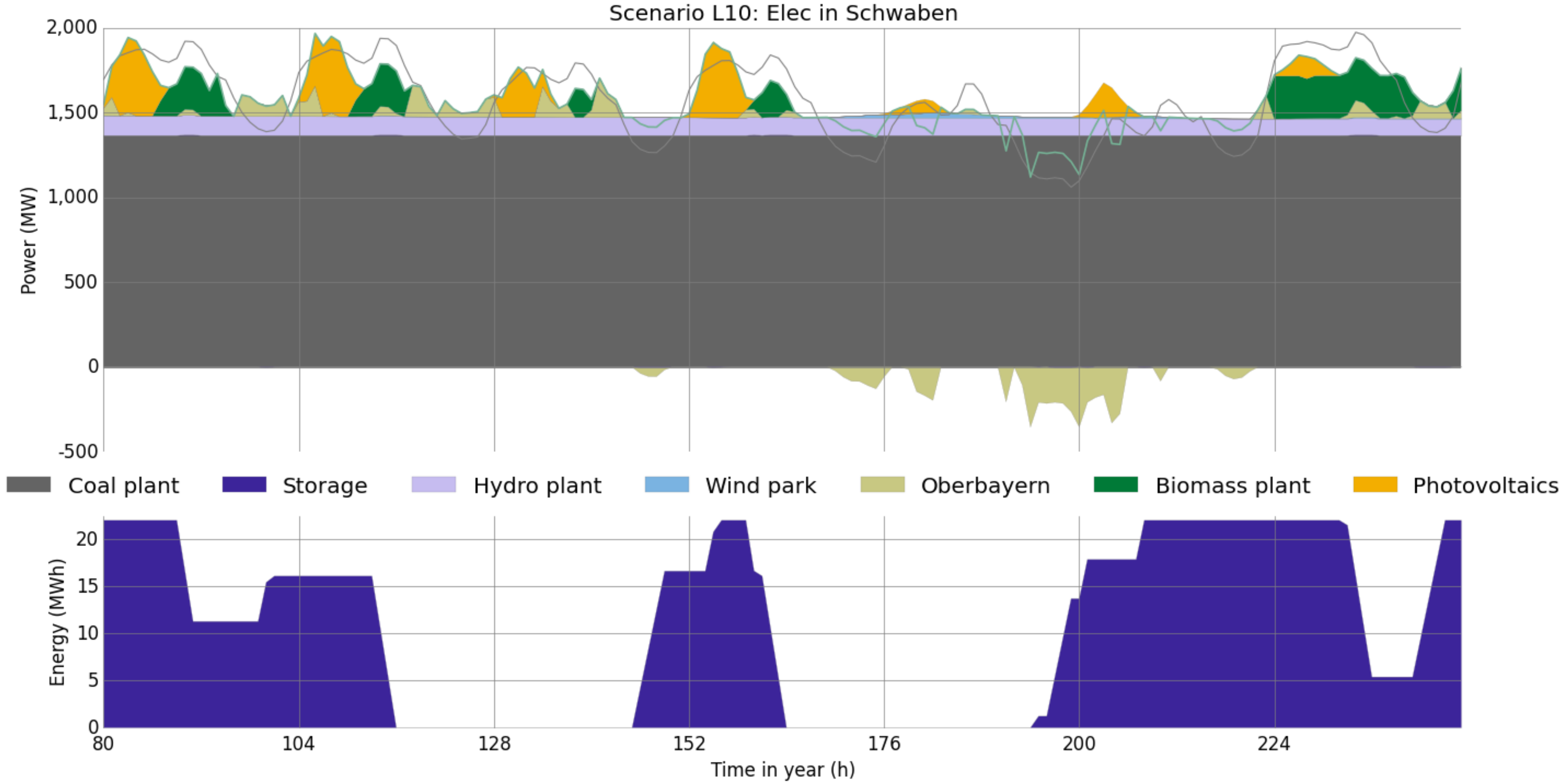


Scenario L10: Elec in Niederbayern
Power (MW)
2,000
1,500
1,000
500
0
-500
-1,000
Coal plant
Wind park
Storage
Mittelfranken
Hydro plant
Oberpfalz
Biomass plant
Oberbayern
Photovoltaics
Energy (MWh)
100
80
60
40
20
0
80
104
128
152
176
200
224
Time in year (h)



Scenario L10: Elec in Oberfranken
Power (MW)
2,000
1,500
1,000
500
0
-500
Coal plant
Oberpfalz
Hydro plant
Unterfranken
Mittelfranken
Photovoltaics
Biomass plant
Wind park
Gud plant
Energy (MWh)
0
80
104
128
152
176
200
224
Time in year (h)


Scenario L10: Elec in Oberpfalz
Power (MW)
1,400
1,200
1,000
800
600
400
200
0
-200
-400
Coal plant
Hydro plant
Storage
Oberfranken
Niederbayern
Wind park
Mittelfranken
Photovoltaics
Biomass plant
Energy (MWh)
400
350
300
250
200
150
100
50
0
80
104
128
152
176
200
224
Time in year (h)


Scenario L10: Elec in Schwaben
Power (MW)
2,000
1,500
1,000
500
0
-500
Coal plant
Storage
Hydro plant
Wind park
Oberbayern
Biomass plant
Photovoltaics
Energy (MWh)
20
15
10
5
0
80
104
128
152
176
200
224
Time in year (h)

Bibliography

[1] *The End Of Fossil Fuels*. Ecotricity Group Limited. URL: https://www.ecotricity.co.uk/our-green-energy/energy-independence/the-end-of-fossil-fuels (cit. on p. 7).

[2] Jacopo Torriti. *Peak Energy Demand and Demand Side Response*. Routledge, 2016. URL: https://www.routledge.com/products/9781138016255 (cit. on p. 7).

[3] Schill W-P Zerrahn A. "On the representation of demand-side management in power system models". In: *Energy* 84 (2015), pp. 840–845. URL: http://www.sciencedirect.com/science/article/pii/S036054421500331X (cit. on pp. 7, 8, 10–13, 16–18).

[4] Government of Western Australia Office of Energy. *Energy Conservation and Demand Management Program*. Dec. 2010. URL: http://www.finance.wa.gov.au/cms/Public_Utilities_Office/Energy_Initiatives/Residential_building_energy_efficiency.aspx (cit. on p. 7).

[5] Danny Bradbury. *Volatile energy prices demand new form of management*. Nov. 2007. URL: http://www.businessgreen.com/bg/analysis/1804282/volatile-energy-prices-demand-form-management (cit. on p. 7).

[6] Jeff St. John. "California Sets the Stage for Radical Grid-Edge Integration". In: *Green Tech Media* (Aug. 2015). URL: http://www.greentechmedia.com/articles/read/california-sets-the-stage-for-radical-grid-edge-integration (cit. on p. 7).

[7] Johannes Dorfner. Python. Institute for Renewable and Sustainable Energy Systems. July 2015. URL: https://github.com/tum-ens/urbs (cit. on pp. 8, 15).

[8] Göransson L et al. "Linkages between demand-side management and congestion in the European electricity transmission system". In: *Energy* 69 (2014), pp. 860–872. URL: http://www.sciencedirect.com/science/article/pii/S0360544214003478 (cit. on pp. 8, 10, 12, 13).

[9] Hunt Allcott. "The Smart Grid, Entry, and Imperfect Competition in Electricity Markets". In: *National Bureau of Economic Research Working Paper* 18071 (May 2012). URL: http://www.nber.org/papers/w18071 (cit. on p. 9).

[10] Severin Borenstein. "The Long-Run Efficiency of Real-Time Electricity Pricing". In: *Energy Journal* 26.3 (2005), pp. 93–116. URL: http://www.iaee.org/en/publications/ejarticle.aspx?id=2101 (cit. on p. 9).

[11] H.V. Poor Wei-Yu Chiu Hongjian Sun. “Demand-side energy storage system management in smart grid”. In: *Third International Conference on Smart Grid Communications (SmartGridComm)*. IEEE. Nov. 2012, pp. 73–78 (cit. on p. 9).

[12] Alizadeh M-I. Aghaei J. “Demand response in smart electricity grids equipped with renewable energy sources”. In: *Renewable and Sustainable Energy Reviews* 18 (2013), pp. 64–72. URL: http://www.sciencedirect.com/science/article/pii/S1364032112005205 (cit. on p. 9).

[13] Annette Lang Dr. Matthias Lang. “Overview Renewable Energy Sources Act”. In: *German Energy Blog* (2014). URL: http://www.germanenergyblog.de/?page_id=283 (cit. on p. 9).

[14] Frieder Borggrefe Moritz Paulus. “The potential of demand-side management in energy-intensive industries for electricity markets in Germany”. In: *Applied Energy* 88.2 (Feb. 2011), pp. 432–441. URL: http://www.sciencedirect.com/science/article/pii/S0306261910000814 (cit. on p. 9).