Disaggregation of energy system optimization models using machine learning for identification of active constraints

David Cardona-Vasquez[a,b,*], Alexander M. Konrad[a], Yannick Werner[a], Sonja Wogrin[a,b]

[a] *Institute of Electricity Economics and Energy Innovation (IEE), Graz University of Technology, Inffeldgasse 18, Graz, Austria*
[b] *Research Center ENERGETIC, Graz University of Technology, Rechbauerstraße 12, Graz, Austria*

1. Appendix

In this Appendix, we provide an additional description of the model used to represent the Austrian power system for the test case and a more in-depth analysis of the machine learning training process; finally, we expand the results concerning the comparison between the complete and disaggregated optimization model results.

1.1. Representation of the Austrian Power System

In our representation of the Austrian power system, we use a 10-node transport model (i.e., pure network flow) with fixed transport costs. This representation allows us to isolate the effect of the time-linking constraints and leverage the Basis-Oriented approach's theoretical foundation. Concerning the imports and exports, and given the geographical location of Austria, it is important to take into account its interconnections with other countries. To achieve this, we divide the neighboring countries into net exporters to Austria (imports) and importers from Austria (exports), where the exporters are represented as a power plant with an installed capacity up to the interconnection's limit, and the importers are modeled as an additional node with a demand corresponding to the historical exports data.

We also subtract long-duration storage from the whole power system demand using historical data, as is commonly done in some ESOM. As mentioned in the main part of the manuscript, long-duration storage leads to extended linked periods,

*Corresponding author:
URL: david.cardonavasquez@tugraz.at (David Cardona-Vasquez)

which hinder the methodology's computational advantage. Dealing with this problem is part of our future research.

With these considerations in mind, the complete model formulation can be found in Eq (1) and Table 1 describes the symbols used in the model.

$$\min_{p_{g,k}, nsp_{i,k}, f_{k,i,j}} \sum_{k,g} C^g p_{g,k} + \sum_{k,i} C^{nsp} nsp_{i,k} + \sum_{k,i,j} C^N f_{k,j,i} \tag{1a}$$

$$\text{s.t.} \quad \sum_j f_{k,j,i} - \sum_j f_{k,i,j} + nsp_{i,k} + \sum_{g \in i} p_{g,k} = D_{k,i} \quad \forall k, i \tag{1b}$$

$$0 \leq p_{w,k} \leq CF_k \overline{P_g} \quad \forall k, w \tag{1c}$$

$$0 \leq p_{t,k} \leq \overline{P_t} \quad \forall k, t \tag{1d}$$

$$f_{k,i,j} \leq \overline{F_l} \quad \forall k, i, j \tag{1e}$$

$$f_{k,j,i} \leq \overline{F_l} \quad \forall k, i, j \tag{1f}$$

$$p_{t,k} - p_{t,k-1} \leq RU_t \quad \forall k, t \tag{1g}$$

$$p_{t,k-1} - p_{t,k} \leq RD_t \quad \forall k, t \tag{1h}$$

$$\tag{1i}$$

Table 1: Description of the symbols used in the energy system optimization model

Symbol	Type	Description
g	Index set	Represents all the generation resources
t	Index set	Represents the ramp-constrained generation resources
w	Index set	Represents the generation resources that are constrained by a capacity factor
i, j	Index set	Represent the nodes
k	Index set	Represents the time index
C^g, C^{nsp}, C^N	Input data	Represent, respectively, the cost coefficients of the energy production the non-supplied power at each, and the transport
$\overline{P_g}, \overline{P_g}$	Input data	Represents the upper limit for generation resources
$\overline{F_l}$	Input data	Represents the maximum transport capacity between the nodes
RU_t, RD_t	Input data	Represent, respectively, the ramp-up and ramp-down limits of ramp-constrained generation resources
$f_{k,i,j}$	Variable	Represents the flow from i to j at time k
$p_{g,k}$	Variable	Represents the generation of resource g at time k
$nsp_{i,k}$	Variable	Represents the non-supplied power at node i and time k

1.2. Classifier Training and Validation Details

In the classifier selection and training process we used a two-step approach, using a low-variance, high-bias classifier to identify relevant features, and then a high-variance, low-bias for the actual prediction. We did it this way because we wanted to use features which can be explained and interpreted from a modeler perspective, and not only based on the improvements they achieve on the accuracy of the model; the complete list of candidate features is presented in Table 2, the *Importance* column

corresponds to the absolute relative importance the logistic regression assigned to the feature.

Table 2: Feature importance according to the coefficients from the logistic regression

Feature Name	**Importance**	**Feature Name**	**Importance**
ren_demand_ratio	0.134	*net_demand*	0.020
ren_demand_ratio_lag_1	0.133	*net_demand_lag_3*	0.019
ren_demand_ratio_lead_1	0.092	*max_gen_lag_3*	0.018
ren_demand_ratio_lag_2	0.071	*max_gen*	0.015
ren_demand_ratio_lag_3	0.067	*max_gen_lag_2*	0.015
demand_lag_2	0.049	*net_demand_lead_2*	0.014
max_gen_lead_3	0.046	*ren_demand_ratio_lead_3*	0.013
max_gen_lead_2	0.036	*max_gen_exp_cap_ratio*	0.008
net_demand_lag_2	0.034	*max_gen_exp_cap_ratio_lag_1*	0.007
max_gen_exp_cap_ratio_lead_1	0.030	*demand_lag_1*	0.007
max_gen_lead_1	0.029	*net_demand_lead_1*	0.006
demand	0.027	*net_demand_lag_1*	0.005
demand_lead_1	0.027	*demand_lead_2*	0.005
net_demand_lead_3	0.026	*ren_demand_ratio_lead_2*	0.003
demand_lag_3	0.022	*max_gen_lag_1*	0.002
demand_lead_3	0.021		

For the low-variance classifier we used a logistic regression, a widely used and tested tool in data analysis, while for the actual predictions we tried several models: support vector with linear and radial basis kernels, nearest neighbors, Ada boost and random forests. The results show that the performance of Ada boost and random forests was basically indistinguishable from one another, nevertheless we chose random forests because they allow for easier parallelization, thus are faster to train in a multi-processor setup like one with GPUs.

Table 3: Performance of other classifiers in the testing portion of the 2022 dataset

Algorithm	**Accuracy**	**Balanced Accuracy**
SVC linear	0.81	0.68
SVC deg.=2	0.81	0.69
SVC deg.=3	0.81	0.67
Random Forest	0.93	0.89
Ada Boost	0.93	0.89

Finally, in Table 4 we present the confusion matrices in relative terms for the years 2015 through 2021 for the classifier without any re-training. As we can see, the

performance of the classifier deteriorates the further we go back in time as shown by the increase of false positives.

Table 4: Performance of the classifier for additional years without re-training

	Predicted							
	2021		**2020**		**2019**		**2018**	
	Linked	*Unlinked*	*Linked*	*Unlinked*	*Linked*	*Unlinked*	*Linked*	*Unlinked*
Linked	0.44	0.08	0.48	0.08	0.40	0.10	0.35	0.10
Unlinked	0.01	0.46	0.01	0.43	0.01	0.49	0.02	0.53
	2017		**2016**		**2015**			
	Linked	*Unlinked*	*Linked*	*Unlinked*	*Linked*	*Unlinked*		
Linked	0.40	0.09	0.38	0.09	0.20	0.16		
Unlinked	0.02	0.49	0.01	0.52	0.03	0.61		

1.3. Additional Results

Let us further discuss the solving times of the individual submodels. First, the submodels with a length longer than 1 hour have an average solving time of 0.07 s, with a maximum of 0.4 s for a submodel of length 70. As for the submodels of 1-hour length, the average solution time was 0.01 seconds. In Fig 1 we see that, as expected, submodels with longer time horizons take longer to solve, with an apparent linear relationship.

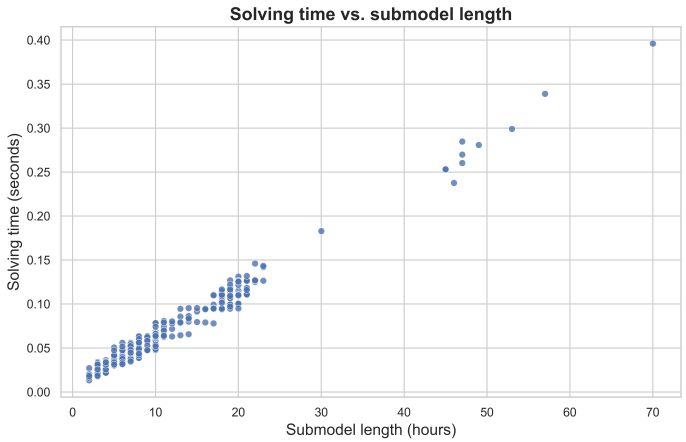

Figure 1: Solving time vs. submodel length, longer submodels took longer to solve

In Fig 2 we can see the distribution of solving time taking into account the submodel length; this result is consistent with the previous figure; however, in this

case we can also see how much more frequent smaller submodels are with respect to larger ones, this allows us to highlight the potential computational gains if one can correctly identify this submodels in the time index.

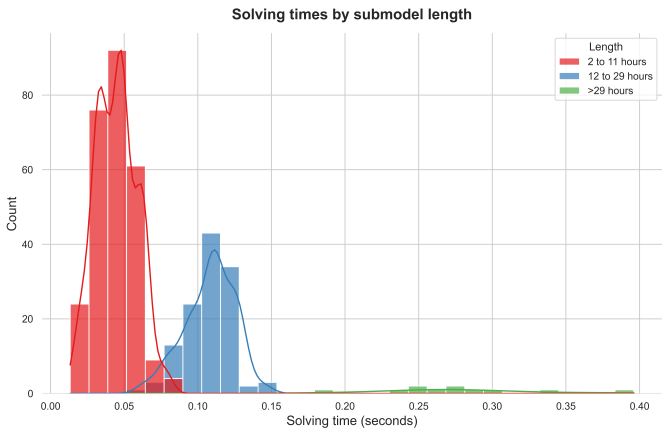

Figure 2: Distribution of solving time by model length, longer models are highly infrequent

Finally, in Table 5 we can see the aggregated production values for each of the generation resources in our representation of the Austrian power system. The values correspond to the validation year (2023) and the *Error* is calculated as the absolute value of the relative error.

Generation Resource	Aggregated Model (MWh)	Complete Model (MWh)	Error (%)
BGLONSHORE	2,378,723.68	2,378,723.68	-
BGLPV	732,351.16	732,351.16	-
BGLROR	5,649.07	5,659.77	0.19
KTNONSHORE	16,210.54	16,210.54	-
KTNPV	645,040.59	645,040.59	-
KTNROR	3,014,895.81	3,014,942.30	0.00
KTNTKWCNG	196,877.13	195,601.01	0.65
NOECZWCNG	-	-	-
NOEONSHORE	2,502,945.92	2,502,945.92	-
NOEPV	1,636,761.09	1,636,564.74	0.01
NOEROR	6,231,501.71	6,231,478.16	0.00
NOETKWCNG	869,256.29	874,173.70	0.56
OOEDEWCNG	-	-	-
OOEONSHORE	55,246.57	55,246.57	-
OOEPV	2,327,632.49	2,327,632.49	-
OOEROR	10,553,935.10	10,553,935.10	-
OOETKWCNG	57,496.88	57,861.45	0.63
OTRONSHORE	-	-	-
OTRPV	43,746.04	43,746.04	-
OTRROR	815,809.82	815,809.82	-
STMKONSHORE	366,935.34	366,935.34	-
STMKPV	1,446,430.86	1,446,785.26	0.02
STMKROR	3,229,830.64	3,230,062.87	0.01
STMKTKWCNG	5,432,464.97	5,424,945.25	0.14
SZBGONSHORE	-	-	-
SZBGPV	474,389.49	474,386.27	0.00
SZBGROR	1,706,265.09	1,706,288.63	0.00
SZBGTKWCNG	3,260.06	3,260.06	-
TIRCHWCNG	2,054.35	2,498.60	17.78
TIRDEWCNG	-	-	-
TIRONSHORE	-	-	-
TIRPV	408,857.22	408,857.22	-
TIRROR	2,831,743.16	2,831,743.16	-
VBGCHWCNG	-	-	-
VBGDEWCNG	-	-	-
VBGONSHORE	-	-	-
VBGPV	388,673.96	388,673.96	-
VBGROR	525,675.02	525,675.02	-
WIEONSHORE	13,420.09	13,420.09	-
WIEPV	429,295.97	429,295.97	-
WIEROR	920,567.19	920,567.19	-
WIETKWCNG	1,073,668.63	1,076,293.98	0.24

Table 5: Comparison of the decision variables from the disaggregated and complete model runs for each generation resource.