Identifying critical demand periods in capacity planning for networks including storage

Andreas Bley and Philipp Hahn

Universität Kassel, Institut für Mathematik, Heinrich-Plett-Straße 40, 34132 Kassel, Germany, andreas.bley@uni-kassel.de, philipp.hahn@uni-kassel.de

Abstract. We consider a capacity planning problem for networks including storage. Given a graph and a time series of demands and supplies, we seek for integer link and storage capacities that permit a single commodity flow with valid storage in- and outtakes over all time steps. This problem arises, for example, in power systems planning, where storage can be used to buffer peaks of varying supplies and demands. For typical time series spanning a full year at hourly resolution, this leads to huge optimization models. To reduce the model size, time series aggregation is commonly used. The time horizon is sliced into fixed size periods, e.g. days or weeks, a small set of representative periods is chosen via clustering methods, and a much smaller model involving only the chosen periods is solved. Representative periods, however, typically do not contain the situations with the most extreme demands and supplies and the strongest effects on storage.
In this paper, we show how to identify such critical periods using principal component analysis (PCA) and convex hull computations and we compare the quality and solution time of the reduced models to the original ones for benchmark instances derived from power systems planning.

Keywords: Time Series Analysis, Capacity Planning, Model Reduction

1 Introduction

We consider a single-commodity capacitated network design problem with storage, which arises, for example, in the planning of power and transport networks where storage can be used to balance loads among consecutive load scenarios. Given a graph and a time series of load scenarios, the task is to find minimum cost edge and storage capacities that permit single-commodity flows in each time step without exceeding the edge and storage capacities. For instances with many time steps, the resulting models are huge and techniques to reduce their size are needed, c.f. [6]. As storage adds dependencies among consecutive time steps, reductions based on isolated time steps do not work well. Instead, short periods of consecutive time steps need to be considered. In practice, typically few representative periods are chosen via clustering techniques and then a model involving only these is solved [4, 6, 10]. Representative periods, however, often do not cover the extreme scenarios governing the capacity installation. In this paper,

we extend the technique presented in [1], which is based on principal component analysis and convex hull computations, to find such extreme periods.

Formally, in the (single-commodity) capacitated network design problem with storage (CNDS) we are given a graph $G = (V, E)$ and, for each edge $ij \in E$, a capacity unit u_{ij}, that can be installed in integer multiples at cost c_{ij} per unit. Storage capacity can be installed in integer multiples of u_i at cost c_i at each node $i \in V_S \subseteq V$. Let $A = \{(i,j), (j,i) \mid ij \in E\}$. All arcs in A except for a subset $A^0 \subset A$ can carry flow. Furthermore, we are given a collection of time periods, each containing several consecutive time steps. Time-dependencies exist only between time steps within the same period. Using this scheme, a full time series is given as a single period containing all time steps, while a reduced model contains only few short periods taken from the time series. We let $[k] := \{1, \dots, k\}$ for each $k \in \mathbb{N}$. We denote by P the number of given periods and, for each $p \in [P]$, by S_p the number of time steps in period p. The set of all time steps is $T = \{(p,s) \mid p \in [P], s \in [S_p]\}$. Let $T^0 = T \cup \{(p,0) \mid p \in [P]\}$. For each time step $(p,s) \in T$, we are given a vector $d^{p,s} = (d_j^{p,s})_{j \in V}$, where $d_j^{p,s}$ is the supply or demand at node j in time step (p,s). Our goal is to find minimum cost edge and storage capacities, such that, for each $(p,s) \in T$, there are storage in- and outtakes and a flow that respect the storage and edge capacities and satisfy the resulting node balances.

Using variables $x_{ij}, y_i \in \mathbb{Z}_+$ for the number of capacity and storage units installed on edge $ij \in E$ and node $i \in V$, variables $l_i^{p,s,+}$, $l_i^{p,s,-}$, $\bar{l}_i^{p,s}$, $\bar{l}_i^{p,0} \in \mathbb{R}_+$ for the storage in- and outtake and level at node $i \in V$ in time step $(p,s) \in T$, and variable $f_{(i,j)}^{p,s} \in \mathbb{R}_+$ for the flow sent via arc $(i,j) \in A$ in time step $(p,s) \in T$, we obtain the following MILP-model for CNDS:

$$\min \quad \sum_{ij \in E} c_{ij} x_{ij} + \sum_{i \in V} c_i y_i \qquad \text{(CNDS-IP)}$$

$$\text{s.t.} \quad f_{(i,j)}^{p,s} + f_{(j,i)}^{p,s} \le x_{ij} u_{ij} \qquad ij \in E,\ (p,s) \in T \qquad (1)$$

$$\bar{l}_i^{p,s} \le y_i u_i \qquad i \in V,\ (p,s) \in T^0 \qquad (2)$$

$$l_i^{p,s,+} - l_i^{p,s,-} + \bar{l}_i^{p,s-1} = \bar{l}_i^{p,s} \qquad i \in V_S,\ (p,s) \in T \qquad (3)$$

$$\sum_{(j,i) \in \delta^-(i)} f_{(j,i)}^{p,s} - \sum_{(i,j) \in \delta^+(i)} f_{(i,j)}^{p,s} - l_i^{p,s,+} + l_i^{p,s,-} = d_i^{p,s} \qquad i \in V,\ (p,s) \in T \qquad (4)$$

$$f_{(i,j)}^{p,s} = 0 \qquad (i,j) \in A_0,\ (p,s) \in T \qquad (5)$$

$$l_i^{p,s,+}, l_i^{p,s,-}, \bar{l}_i^{p,s} = 0 \qquad i \in V \setminus V_S,\ (p,s) \in T \qquad (6)$$

$$l_i^{p,s,+}, l_i^{p,s,-}, \bar{l}_i^{p,s} \ge 0 \qquad i \in V,\ (p,s) \in T$$

$$f_{(i,j)}^{p,s} \ge 0 \qquad (i,j) \in A,\ (p,s) \in T$$

$$x_{ij} \ge 0, \quad x_{ij} \in \mathbb{Z} \qquad ij \in E$$

$$y_i \ge 0, \quad y_i \in \mathbb{Z} \qquad i \in V_S$$

Inequalities (1) and (2) ensure that in each time step the flow does not exceed the edge capacities and the storage levels do not exceed storage capacities. Constraints (3) link the storage levels of consecutive time steps within a period

and the corresponding in- and outtakes. Equalities (4) ensure that the flow in each time step satisfies all node balances including storage in- and outtakes. For simplicity, flow and storage variables are defined for all arcs and nodes but fixed to zero for all non-flow arcs and non-storage nodes in (5) and (6), respectively. The objective is to minimize the sum of all capacity installation costs.

Note that the capacity variables x_{ij} and y_i do not depend on the time steps, while flow and storage variables do. Also note that the initial storage level in each period is unrestricted. Thus, initially full storage can be used to help satisfy demands. The storage capacity needed for this, however, causes costs.

2 Reducing the model

To model the CNDS problem for a full time series and all dependencies between consecutive time steps, a single period containing all time steps is given as input to the model (CNDS-IP). Our goal is to construct a smaller, approximate model that involves only few shorter periods and less time steps in total. For this, we first generate a large collection of short periods that cover the original time series and then pick a small sub-collection of these to remain in the reduced model.

2.1 Generating periods

Assume the full time series is given as a single period containing all m time steps and we are given a desired sub-period length S, e.g. $S = 168$ for a period of 1 week and hourly time steps, and a desired shift Q between two consecutive periods, e.g. $Q = 24$ for a shift of 1 day. For simplicity, assume that $P = (m - S)/Q \in \mathbb{N}$. We then construct P sub-periods of size S, each shifted by a multiple of Q. More precisely, the S consecutive time steps $(p-1)Q+1$ to $(p-1)Q+S$ of the original full time series are contained in period $p \in [P]$ and we have $d^{p,s} = d_{\text{orig}}^{1,(p-1)Q+s}$, where d_{orig} denotes the demand vectors indexed according to the original full time series and d the demand (re-)indexed according to the time steps s in the created period $p \in [P]$.

Passing the periods in $[P]$ (or a sub-collection thereof) with the corresponding demand vectors d as input to (CNDS-IP), one obtains a relaxation of the model for the full time series. Edge and storage capacities that permit valid flows and storage in- and outtakes for the full times series' model also permit valid flows and storage in- and outtakes for the model involving the periods in $[P]$ instead.

2.2 Identifying critical periods

In order to identify a small sub-collection of the generated periods, we extend the method presented in [1] to select critical isolated scenarios in robust network design (without dependencies among time steps) to periods spanning multiple time steps. That method first uses principal component analysis (PCA) to identify the directions of the statistically largest variation among the demand vectors. In the second step, all demand vectors are projected onto subspaces spanned by

one or more of the most important principal components. For each of these subspaces, the vertex set of the convex hull covering the projected demand vectors is computed. Eventually, the scenarios corresponding to the vertices are selected as critical and chosen to remain in the reduced problem. In a symmetrized variant of this method, for each scenario s both the original demand vector d^s and its negative $-d^s$ are projected onto the subspaces, which may lead to fewer scenarios corresponding to vertices. For subspace dimension b, we denote the non-symmetrized and the symmetrized variants by CH_b and SCH_b, respectively.

To extend that method to time periods, we first apply a slicing procedure to the original demand vectors in order to create a new matrix whose rows represent the demands of periods of length S. More precisely, with $n = |V|$, the given node demands form a matrix $D_{\text{orig}} \in \mathbb{R}^{m \times n}$, whose s-th row is the demand vector $d^{1,s}_{\text{orig}}$ for time step s. Concatenating the demand vectors $d^{p,s}$ of all time steps $s \in [S_p]$ in a sub-period $p \in [P]$ to a single row, we obtain the matrix $D \in \mathbb{R}^{P \times S \cdot n}$. The p-th row D_p of D can be regarded as the demand or 'feature' vector of the entire period p. Eventually, we apply the methods described in [1] to the feature vectors D_p for periods $p \in [P]$ to choose a small subset of extreme periods $P' \subseteq [P]$, that will be finally kept in the reduced model.

Time series, however, typically feature some natural periodicities, such as daily and weekly patterns. If the shift Q is no multiple of these periodicities, seemingly strong variations among different periods' demands are artificially introduced by the different offsets. To eliminate these artificial variations, we can realign the periods' demand data to a uniform pattern before applying the PCA. For weekly periods, for example, we cyclically reorder the time steps $s \in [S_p]$ in each row p in such a way that the time step corresponding to Sunday 0:00 is in the same column in each row.

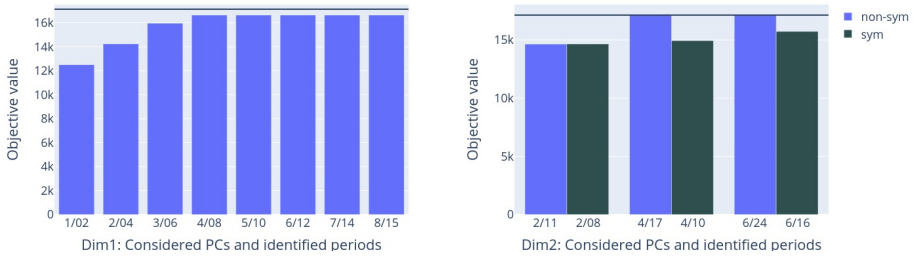
Note that, in contrast to the approach presented in [2], our approach does not require to first solve (CNDS-IP) for each period $p \in [P]$ in order to decide which ones to use in the reduced model. It solely relies on analyzing the extremality of the (adjusted) demand vectors, which is computationally much less demanding.

3 Instances and Computational Experiments

To assess the effectiveness of our approach, we compare the solution times and the objective values of (CNDS-IP) for the full time series and for the chosen subset of periods for some benchmark instances. The method is implemented in Python 3.7. We use Gurobi 9.5.0 [3] as ILP solver, except for the Benders decomposition approach, where CPLEX 12.9 [5] with its built-in Benders algorithm is used. In both cases, we admit a mip-gap of 0.1% to keep solution times relatively small. The reported results are obtained on a machine with two 14-core Intel Xeon (R) E5-2690 v4, 2.6 GHz processors and 256 GB of RAM. The presented plots are created using grblogtools [8] and plotly [9].

In the following, we report on two relatively small benchmark instances derived from a power grid planning problem including storage. In both instances, the network consists of a 6 node densely meshed bidirectional core, resembling

Fig. 1. Objective of reduced models for varying number of PCs for instance I1.

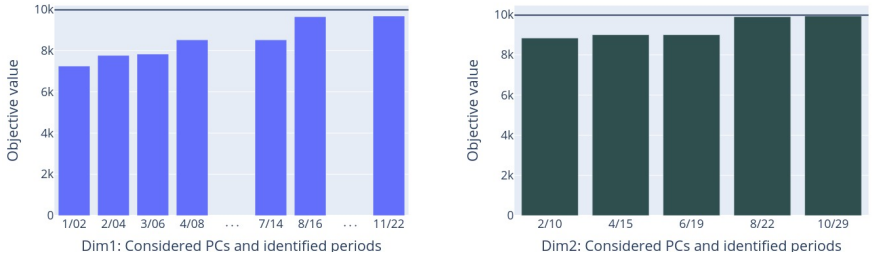


countries and their exchange capacities, and 50 nodes representing 26 renewable and 24 thermal generators, which are unidirectionally connected to only one country node and an artificial sink each. The demands and supplies of the renewable generators and countries are given as a time series, while the thermal generators are flexible within their capacity bounds. The time series of the smaller instance I1 contains 8,736 hourly time steps covering one year, that of instance I2 a total of 61,320 covering seven years. We generate periods of length 168 (1 week) and step size 168, so periods do not overlap. In the PCA-based period selection from section 2.2, we always standardize the data prior applying PCA. This turned out to yield better results than using the unscaled demand values, contrarily to the results obtained in [1] for critical scenarios in robust network design. The reasons are not yet clear to us.

Figure 1 shows the objective values obtained with the subset of periods chosen with our approach depending on the number of principle components (PCs) considered for the smaller instance I1. The left plot shows the results for method variant CH_1, which just chooses the two extreme periods per PC. The right plot shows the results for variants CH_2 and SCH_2. The horizontal line shows the objective for the full time series. Method CH_1 yields a very good approximation already for 4 PCs, choosing 8 periods. Unfortunately, considering more PCs with this variant does not close the remaining gap. Considering the first 4 PCs, variant CH_2 identifies 17 (out of 52) periods, leading to a reduced model that actually achieves the full model's optimal value. With the symmetric variant we miss one of the critical periods. The model for the full time series needs 11,011s to be solved to optimality, a gap of 0.1% is reached after 2,180s. The solution times for the reduced models range from 10s for 2 periods to roughly 250s for 17 and 500s for 24 periods. The generation and selection of the periods, including PCA and convex hull computations, requires just a few seconds.

The results for instance I2 are shown in Figure 2. For this instance, the full time series model could not be solved in reasonable time. After 14,881s the computation was aborted with a gap of 0.12%. Again, the non-symmetric variant for dimension 1 is shown left and the symmetric variant for dimension 2 right. As before, we obtain a very good approximation with variant CH_1, choosing only 16 out of 365 periods when considering the first 8 PCs, but we cannot close the remaining gap. For CH_2, this gap closes almost with the first 6 and completely

Fig. 2. Objective of reduced models for varying number of PCs for instance I2



with the first 8 PCs considered, leading to 22 or 29 out of 365 periods chosen, respectively. The runtimes of the reduced models range from 17s for 2 to roughly 1000 seconds for 29 considered periods.

Using Benders algorithm with the 22 and 17 identified critical periods in the initial master problem and generating Benders cuts for the others, we solve the full models of I2 and I1 involving all 61,320 and 8,736 time steps in 11,740s and 3,770s while a gap of 0.1% is reached after 9,400s and 1,200s, respectively.

4 Conclusion

In this paper, the method from [1] for identifying critical demand scenarios has been extended to critical demand periods spanning multiple time steps. The method solely works on the raw demand data. Neither the topology, the capacity, nor any other structural properties are considered. Also, the method is very fast. In relation to solving (the reduced) CNDS-IP the times for PCA and convex hull computations are negligible. In our experiments, the resulting reduced models deliver good approximations of the original ones. Further, using the critical periods in the master problem of a Benders decomposition leads to a significant speed up for finding an exact solution. In the future, we plan to apply this method to a large scale sector coupled energy systems planning model where the model is reduced using both representative and critical periods.

Acknowledgements This work was supported by the German Federal Ministry for Economic Affairs and Energy (BMWi), Project *NSON II*, [7].

This version of the contribution has been accepted for publication after peer review, but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at:
`https://dx.doi.org/10.1007/978-3-031-24907-5_27`.

References

1. Bley, A., Hahn, P.: Identifying critical demand scenarios for the robust capacitated network design problem using principal component analysis. Tech. rep., Institute for Mathematics, University of Kassel (2021)
2. Bukenberger, J.P., Webster, M.D.: Approximate latent factor algorithm for scenario selection and weighting in transmission expansion planning. IEEE Transactions on Power Systems 35(2), 1099–1108 (2020)
3. Gurobi Optimization, LLC: Gurobi Optimizer Reference Manual (2022), `https://www.gurobi.com`
4. Hoffmann, M., Kotzur, L., Stolten, D., Robinius, M.: A review on time series aggregation methods for energy system models. Energies 13, 641 (2020)
5. IBM ILOG CPLEX: User's manual for CPLEX 12.9 (2020)
6. Mahdavi, M., Antunez, C.S., Ajalli, M., Romero, R.: Transmission expansion planning: Literature review and classification. IEEE Systems Journal 13(3), 3129–3140 (2019)
7. Mende, D., Harms, Y., Härtel, P., Frischmuth, F., Stock, D.S., Braun, M., Herrmann, M., Hofmann, L., Valois, M., Bley, A., Hahn, P., Jurczyk, J., Rathke, C.: NSON II: Next steps in economical connection and international integration of offshore wind energy in the north seas. In: Proceedings of the 19th Wind Integration Workshop, Ljubljana, Slovenia (2020)
8. Miltenberger, M., Oberdieck, R., Siefen, K., Aramon, M., Bowly, S.: grblogtools, `https://github.com/Gurobi/grblogtools`
9. Plotly Technologies Inc.: Collaborative data science (2015), `https://plot.ly`
10. Saxena, A., Prasad, M., Gupta, A., Bharill, N., Patel, O.P., Tiwari, A., Er, M.J., Ding, W., Lin, C.T.: A review of clustering techniques and developments. Neurocomputing 267, 664–681 (2017)