Planning macro-energy systems with multiple climatic years - A quadratic trust-region approach for Benders decomposition

Leonard Göke\textsuperscript{a,b,*}, Mario Kendziorski\textsuperscript{a,b} and Felix Schmidt\textsuperscript{b}

\textsuperscript{a}Berlin University of Technology, Workgroup for Infrastructure Policy (WIP), 10623 Berlin, Germany.
\textsuperscript{b}German Institute for Economic Research (DIW Berlin), Department of Energy, Transportation, Environment, 10117 Berlin, Germany.

\begin{abstract}
This paper applies Benders decomposition to two-stage stochastic problems for energy planning with multiple climatic years, a key problem for the design of renewable energy systems. First, we implement Benders decomposition with existing enhancements suited for the characteristics of the problem, a simple continuous master-problem and few but large sub-problems. Next, we develop a novel trust-region method using a quadratic constraint that is continuously adapted to further improve the algorithm.

In a quantitative case-study our method accelerates Benders decomposition by a factor of four to six slightly increasing solve time of the master-problem, but greatly reducing the number of iterations. With the computational resources at our disposal, Benders decomposition with quadratic trust-region outperforms closed optimization if planning covers more than six climatic years, because run-time does not increase with the number of scenarios thanks to distributed computing. Furthermore, results show that the quadratic trust-region approach benefits from a heuristic starting solution but does not depend on it to be performative.

Finally, we suggest further improvements of the algorithm. First, heuristic methods to narrow the solution space of the master-problem. Second, approximations of the sub-problems to faster add inexact but valid cuts.
\end{abstract}

1. Introduction

To achieve objectives of the Paris Climate Agreement and limit global warming, the energy system must undergo a major transformation and replace 86\% of primary energy from fossil fuels as of 2019 with renewable resources (Ritchie & Roser, 2020). At the heart of this transformation is renewable electricity from wind and photovoltaic (PV) for two reasons: First, its technical potential exceeds other renewables and even demand projections (Creutzig et al., 2017). Estimates for the global potential of PV alone range from 1,585 to 50,580 EJ, at least three times 2019’s primary consumption. Second, use of renewable electricity is not limited to the power sector, but can also be deployed for electric heating, mobility, and the creation of synthetic fuels to decarbonize the heat, transport, and industry sector.

1.1. System planning with multiple climatic years

Key tools for analyzing the transformation towards renewable electricity are techno-economic planning models, stylized representations of macro-energy systems describing the flow and conversion of energy. Formulated as linear optimization problems, these models decide from a portfolio of technologies on their expansion and operation to satisfy an exogenous final demand at minimum costs and subject to boundary
Benders decomposition with quadratic trust-region conditions, for instance emission limits. Typical applications include the development of long-term scenarios for the energy system, assessment of technologies in a system context, and analysis of energy policy (Göke, Weibezahn, & von Hirschhausen, 2021).

The fluctuating nature of PV and wind generation is a challenge for planning models. For conventional energy systems characterized by dispatchable generation from thermal power plants, a small number of representative time-periods is sufficient to achieve accurate results. Capturing fluctuations of PV and wind however requires a much higher temporal resolution, that greatly increases model size and easily renders the linear optimization problem computationally intractable (Göke & Kendziorski, 2021). Addressing this issue, different techniques have been proposed to reduce computational complexity while representing renewables accurately, for instance iteratively adjusting the representative time-periods or limiting high resolution to selected parts of the system (Göke, 2021b; Teichgraeber, Küpper, & Brandt, 2021).

Overall efforts to make models computationally tractable and capture fluctuations of renewables are focused on representing a single climatic year accurately, but research finds substantial variability across different climatic years, for instance in renewable generation, that heavily impact model results (Pfenninger, 2017; Pfenninger & Staffell, 2016). Several studies analyze how consideration of multiple climatic years affects planning power systems. Using 40 climatic years, H. C. Bloomfield, Brayshaw, Shaffrey, Coker, and Thornton (2016) investigate effects on the British power system and conclude robust planning should consider 10 climatic years at least. As a key driver of inter-annual variability, they identify the variation of wind generation. Collins, Deane, Ó Gallachóir, Pfenninger, and Staffell (2018) compare the impact of different climatic years on the European power system finding that ambitious expansion of renewables will increase the climate-driven variation of emissions and generation costs by a factor of 5 until 2030 compared to 2015. Ruhnau and Qvist (2022) study the effect of inter-annual variability has on storage requirements for a stylized fully renewable German power system. Compared to a single average climatic year, the amount of energy to be stored more than doubles when considering 35 years of climate data. Beyond the power sector, Lombardi et al. (2022) observe great variations of heating demand across climatic years for Italy. Ohlendorf and Schill (2020) specifically analyze the frequency of low-wind power events in Germany threatening system adequacy, again concluding planning models should consider more climatic years.

Heuristic approaches to consider multiple climatic years are limited. Solving a single-year planning model for various climatic years can indicate the level of variability, but not provide an adequate and cost-efficient solution across all years. Using representative time-periods but base them on multiple climatic years cannot fully capture fluctuations of renewables (Hilbers, Brayshaw, & Gandy, 2019).

A more robust and analytical approach to climatic variability is stochastic programming. Formulated as two-stage problems, planning models decide on the expansion of technologies in the first stage and on their operation in the second. Operation in the second stage is subject to uncertainty represented by scenarios that reflect weather-dependency and are based on distinct year of consistent climate data. On the downside, multiple scenarios, representing one climatic year each, greatly increase model size and aggravate the problem of computational tractability described above. For instance, to enable two-stage planning the EMPIRE model uses representative time-periods and omits seasonal storage, a key component for renewable energy systems (Backe, Skar, del Granado, Turgut, & Tomasgard, 2022).

1.2. Benders decomposition for two-stage stochastic problems

Benders decomposition (BD), first introduced in Benders (1962), is a decomposition technique to solve optimization problems. Van Slyke and Wets (1969) first applied a variation of BD, termed the L-shaped method, to solve two-stage stochastic problems and potentially improve their computational tractability. Applied to energy planning with multiple scenarios, or climatic years, the problem is decomposed into a master-problem (MP) addressing technology expansion, and a number of mutually independent sub-problems (SPs) for operation. Afterwards, the MP and SPs are solved repeatedly to generate constraints, so-called Benders
cuts, that are added to the MP until the algorithm converges (Conejo, Castillo, Minguez, & Garcia-Bertrand, 2006).

In several applications of BD for energy planning models, each SP corresponds to a short time-span of the year, in some models as short as an hour (Brandenberg & Stursberg, 2021; Lohmann & Rebennack, 2017; Skar, Doorman, & Tomaszard, 2014). However, such decomposition is not applicable when modelling high renewable shares and including seasonal storage, which creates dependencies across the entire year. Thus, an accurate representation of these dependencies requires that each SP covers operation for a specific climatic year or scenario instead. Considering the high temporal detail of operation, this results in a simple MP and several large SPs. Furthermore, MP and SPs can contain both discrete and continuous variables, but in the MP at least renewable expansion is continuous considering the size of a single wind turbine or PV panel is small from a macro perspective.

For many problems the original BD converges slowly and is not competitive to closed optimization using off-the-shelf solvers. However, an extensive branch of research proposes different enhancements to improve the original BD. In a review, Rahmaniani, Crainic, Gendreau, and Rei (2017) group them into the following three most important groups:

1. **Solution procedure**: The majority of enhancements in this area reduces the computation time of the MP, like removing non-binding cuts or not solve to optimality in early iterations. The in our case more relevant SPs can deploy distributed parallel computing or sensibly reduce the number of scenarios in early iterations. Furthermore, not solving SPs to optimality can produce inexact but valid cuts (Zakeri, Philpott, & Ryan, 2000).

2. **Solution generation**: Again, most enhancements regarding solution generation aim to accelerate the MP, for instance by solving a linear relaxation instead of a mixed-integer problem. Other adjustments to the MP improve the convergence rate, rendering them sensible even if the MP is simple compared to the SPs. First, multi-cut reformulations using separate cuts for each SP instead of a single aggregated cut improve convergence but make the MP harder. Second, trust-regions restricting the top-problem to solutions close to a reference point, usually the current best solution, avoid heavy oscillation (Ruszczynski, 2003). Santoso, Ahmed, Goetschalckx, and Shapiro (2005), for example, use the $l_0$-norm for binary variables, or Hamming distance, to limit the MP to a radius around the current best solution. Linderoth and Wright (2003) instead apply the continuous $l_\infty$-norm to limit the maximum difference across all variables between a new and the current best solution. Instead of adding constraints, regularized decomposition achieves a stabilizing effect by adding a term to the objective function that penalizes deviations from the reference points, typically using the $l_2$-norm (van Ackooij, Lebbe, & Malick, 2017).

Finally, problem-specific heuristics can provide an initial feasible solution for BD, obtain an upper bound on the objective value, and exclude infeasible solutions at an early stage.

3. **Cut generation**: Magnanti and Wong (1981) note that degenerate SPs, e.g. SPs without unique solutions, can generate a set of different cuts, all affecting Benders convergence differently. To select strong cuts improving convergence, they propose to solve a modified version of the dual SP after the original SP. As a result, the method is most efficient, if the MP is difficult and the SPs are small.

In conclusion, many of the established Benders enhancements aim at problems with a difficult MP and small SPs, typically combinatorial problems with discrete complicating variables. Rahmaniani et al. (2017) correspondingly state "it has often been reported that more than 90% of the total execution [...] time is spent

---

We omitted the fourth group "Decomposition strategies" since none of the few approaches in this group is relevant for us.
on solving the MP”. As a result, applicability to the two-stage planning problem investigated in this paper with a simple MP, but large SPs is limited. In contrast to typical Benders applications, it is not combinatorial complexity but the mere size of the problem that makes it computationally challenging.

1.3. Contribution
In this paper, we apply BD to solve a two-stage stochastic problem with discrete scenarios used for planning renewable macro-energy systems considering multiple climatic years. Our implementation of BD includes a multi-cut reformulation, inexact cuts, parallel computing, and a trust-region paired with a simple heuristic to obtain an initial solution. The contributions of this paper are twofold:

1. We introduce a highly relevant problem with a structure well-suited for BD to the operations research literature and investigate to what extent existing refinements can accelerate the algorithm. Considering many state-of-the-art refinements do not suit the specific structure of the problem, a simple continuous MP and large SPs, we hope to spark further research on this topic.
2. Our Benders implementation deploys a refined version of the trust-region method for continuous variables using the $l_2$-norm and dynamically adapting the region.

The remainder of the paper is structured as follows: The following section 2 generally introduces the investigated planning problem, its BD implementation, and the refined trust-region method. Subsequently, section 3 provides details on the specific planning problem and the approach used for scenario generation that are used in section 4 to benchmark the developed BD implementation. The final section concludes and suggests further developments.

2. Method
The two-stage energy planning problem decides on capacity expansion in the first stage and operation in the second. Operation is subject to uncertainty represented by different scenarios, for instance corresponding to different climatic years. The following subsections first introduce the closed formulation of the problem and then the BD implementations applied to solve it. The notation uses lowercase letters for parameters, uppercase letters for variables, and greek letters for anything specific to BD.

2.1. Two-stage stochastic planning problem
Eqs. 1a to 1j provide the closed formulation of the two-stage stochastic energy planning problem. For a set of technologies $i \in I$, the problem decides on the expansion of capacities $\text{Exp}_{y,i}$ over a set of (not necessarily directly) consecutive years $y \in Y$. All variables of the problem are continuous and non-negative.

According to 1d, the capacity $\text{Capa}_{y,i}$ available in each year, depends on expansion in the current and previous years provided by $Y^\text{Exp}_y$. In Eqs. 1e to 1h these capacities constrain the operation of technologies at each time-step $t \in T$, in each scenario $s \in S$, and within each year. For generation technologies $I^\text{ge}$, the capacity constraint limiting the generation $\text{Gen}_{y,s,i,t}$ includes a scenario-dependent capacity factor $c_{f_{s,i,t}}$ reflecting the available share of capacity. For storage technologies $I^\text{st}$, two different capacities constrain three different operational variables: Charged energy $S_{y,s,i,t}^{\text{in}}$, and discharged energy $S_{y,s,i,t}^{\text{out}}$ are both constrained by the power capacity $\text{Capa}_{y,i}^\text{st}$; the current storage level $S_{y,s,i,t}^{\text{size}}$ is constrained by the energy capacity $\text{Capa}_{y,i}^\text{size}$. In total, net supply from generation and storage technologies must match the exogenous demand $\text{dem}_{y,s,t}$ in each year, time-step, and scenario, as expressed in the energy balance in Eq. 1i. To avoid infeasibility, the balance includes a loss-of-load variable $\text{Lss}_{y,s,t}$. The storage balance in Eq. 1d computes the storage level at time-step $t$ based on the level at the previous time-step $t - 1$ plus charged and minus discharged energy. The storage constraint is circular for each year, meaning the last time-step in a year is previous to the first.

The objective function of the problem in Eq. 1a minimizes total system costs comprised of expansion costs $U$ and the sum of operational costs $V_{y,s}$ across all years $y$ and scenarios $s$ weighted according to the
Benders decomposition with quadratic trust-region

scenario probabilities $p_s$. Expansion costs defined in Eq. 1b depend on the expansion variable and the specific expansion costs $c_{y,s}^{inv}$ of each technology. Operational costs defined in Eq. 1c depend on the costs associated with loss-of-load $c_{y,s,t}^{loss}$ and variable costs of generation technologies $c_{y,s,t}^{var}$, for instance fuel costs.

\[
\begin{align*}
\text{min } U & \quad + \sum_{y \in Y, s \in S} p_s \cdot V_{y,s} \\
\text{s.t. } U & \quad = \sum_{y \in Y, i \in I} c_{y,i}^{inv} \cdot \text{Exp}_{y,i} \\
V_{y,s} & \quad = \sum_{t \in T} L_{y,s,t} \cdot c_{y,s,t}^{loss} + \sum_{i \in I} \text{Gen}_{y,s,i,t} \cdot c_{y,s,t}^{car} \quad \forall y \in Y, s \in S \\
\text{Capa}_{y,i} & \quad = \sum_{y' \in Y_y^p} \text{Exp}_{y',i} \\
\text{Gen}_{y,s,i,t} & \quad \leq c_{f,s,i,t} \cdot \text{Capa}_{y,i}^{ge} \quad \forall y \in Y, s \in S, i \in I^{ge}, t \in T \\
\text{St}_{y,s,i,t}^{out} & \quad \leq \text{Capa}_{y,i}^{st} \quad \forall y \in Y, s \in S, i \in I^{st}, t \in T \\
\text{St}_{y,s,i,t}^{in} & \quad \leq \text{Capa}_{y,i}^{st} \quad \forall y \in Y, s \in S, i \in I^{st}, t \in T \\
\text{St}_{y,s,i,t}^{size} & \quad \leq \text{Capa}_{y,i}^{size} \quad \forall y \in Y, s \in S, i \in I^{st}, t \in T \\
dem_{y,s,t} & \quad = L_{y,s,t} + \sum_{i \in I^{ge}} \text{Gen}_{y,s,i,t} - \sum_{i \in I^{st}} \text{St}_{y,s,i,t}^{in} + \text{St}_{y,s,i,t}^{out} \quad \forall y \in Y, s \in S, t \in T \\
\text{St}_{y,s,i,t}^{size} & \quad = \text{St}_{y,s,i,t}^{size} - 1 + \text{St}_{y,s,i,t}^{in} - \text{St}_{y,s,i,t}^{out} \quad \forall y \in Y, s \in S, i \in I^{st}, t \in T
\end{align*}
\]

The formulation in Eqs. 1a to 1j introduces the general problem structure and all details pivotal for BD, but is still stylized. The model introduced in section 3 and used for benchmarking includes additional elements. First, multiple regions of expansion and operation. Between regions, the model can build transmission capacity in the first stage to exchange energy in the second. Next, distinct energy carriers, like electricity and hydrogen, that are converted into one another by respective technologies, like hydrogen turbines or electrolyzers. Finally, storage losses and additional constraints, imposing upper limits on capacities or restricting operation, for instance a yearly emission limit. For an exhaustive formulation of the planning problem see Göke (2021b).

### 2.2. Benders decomposition of planning problem

Fig. 1 is an illustrative depiction of the planning problem based on the block structure of the constraint matrix. The expansion problem in the first stage includes expansion and capacity variables, the constraint in 1d connecting them and the cost definition in Eq. 1b. The second stage includes the remaining constraints, namely the energy and storage balance, the definition of variable costs, and the capacity restrictions. The latter links capacity with operational variables and connects first and second stage of the problem.

![Figure 1: Structure of closed two-stage problem](image)
Even for a single year and scenario, the number of constraints and variables of the expansion problem in the first stage is small compared to operation in the second. Increasing the number of years extends both stages but increasing the number of stochastic scenarios exclusively extends the operational problem. Since the expansion problem is small, total size almost linearly depends on the number of stochastic scenarios and the problem quickly becomes intractable, if their number increases.

For BD, first stage expansion corresponds to the MP, while operation is split into several independent SPs. For demonstration, consider a problem covering two consecutive years, 2030 and 2040, and two scenarios, representing the climatic years 2000' and 2008'. According to the problem formulation in the previous section, operation is modelled for each year and scenario resulting in four distinct operational problems, as shown in Fig. 2. For a given set of capacities, each operational problem can be solved independently, if no complicating constraint links the variables from different SPs. An example for such a constraint is an emission limit that is not enforced for each SP separately, but jointly across different years and scenarios, like a carbon budget.

Fig. 3 provides the structure of the decomposed problem, analogously to Fig. 1. It shows how the operational problem is split into four independent SPs, each consisting of different operational variables, but sharing capacities with the MP and other SPs for the same year. Splitting second stage operation into several SPs is beneficial, because it enables distributed computing and prevents complexity in the second stage to scale with the number of scenarios. In addition, it increases the number of cuts added to the MP in each iteration improving convergence, as we will elaborate when introducing the BD algorithm next.

Alg. 1 presents the standard version of the BD algorithm applied in this paper. $k$ is an iteration counter; $\xi_{\text{low}}$ and $\xi_{\text{up}}$ are the current lower and upper bound of the objective function, respectively. We use multi-cut and at each iteration add separate constraints for each sub-problem to the MP instead of a single aggregated cut. The approach improves convergence but increases complexity of the MP—a favourable trade-off since the MP is simple. To mitigate adverse effects on MP performance, the algorithm deletes cuts that were not created or binding within the last $\eta$ iterations, with $\eta$ being a pre-set parameter. Furthermore, solving the SPs is fully parallelized using distributed computing.

After initializing all variables, the algorithm starts iterating after line 3 solving the original MP, as described by Eqs. 2a to 2c. Since the MP only includes the constraints of the expansion problem, its objective function approximates the actual costs of each SP $V_{y,s}$ using the estimator $\alpha_{y,s}$. After solving, the algorithm obtains the resulting capacities $Capa_{y,i}^{(k)}$ and expansion costs $U^{(k)}$ for the current iteration and sets the lower bound $\xi_{\text{low}}$ to the current objective of the MP. In the first iteration, $\alpha_{y,s}$ is still unconstrained and the trivial...
optimum for the MP is zero meaning no capacity expansion at all.

\[
\min U + \sum_{y \in Y, s \in S} p_s \cdot \alpha_{y,s} \tag{2a}
\]

\[
\text{s.t. } U = \sum_{y \in Y} c^{\text{inv}}_{y,i} \cdot \text{Exp}_{y,i} \tag{2b}
\]

\[
\text{Capa}_{y,i} = \sum_{y' \in Y_{y,s}} \text{Exp}_{y',i} \quad \forall y \in Y, i \in I \tag{2c}
\]

Next, the algorithm solves each SP in parallel with capacities fixed to the previously computed \( \text{Capa}^{(k)}_{y,i} \), as described by Eqs. 3a to 3i. Afterwards, the algorithm gets the actual costs of each SP \( V^{(k)}_{y,s} \) and the dual values \( \lambda^{(k)}_{y,i} \) of the constraint fixing capacities in Eq. 3i. Note that the SPs are always feasible in our case since they include the a loss-of-load variable \( Lss_{y,s,t} \).

\[
\min V^{(k)}_{y,s} = Lss_{y,s,t} \cdot c^{\text{ls}} + \sum_{i \in I, t \in T} \text{Gen}_{y,s,i,t} \cdot e^{\text{car}}_{y,s,i} \quad \forall y \in Y, s \in S \tag{3a}
\]

\[
\text{s.t. } V^{(k)}_{y,s} \leq c^{\text{f}}_{y,s,i,t} \cdot \text{Capa}_{y,s}^{\text{ge}} \quad \forall y \in Y, s \in S, i \in I^{\text{ge}}, t \in T \tag{3b}
\]

\[
\text{Gen}_{y,s,i,t} \leq \text{Capa}_{y,s}^{\text{st}} \quad \forall y \in Y, s \in S, i \in I^{\text{st}}, t \in T \tag{3c}
\]

\[
\text{St}_{y,s,i,t}^{\text{in}} \leq \text{Capa}_{y,s}^{\text{st}} \quad \forall y \in Y, s \in S, i \in I^{\text{st}}, t \in T \tag{3d}
\]

\[
\text{St}_{y,s,i,t}^{\text{size}} \leq \text{Capa}_{y,s}^{\text{size}} \quad \forall y \in Y, s \in S, i \in I^{\text{st}}, t \in T \tag{3e}
\]

\[
\text{dem}_{y,s,t} = Lss_{y,s,t} + \sum_{i \in I^{\text{ge}}} \text{Gen}_{y,s,i,t} - \sum_{i \in I^{\text{st}}} \text{St}_{y,s,i,t}^{\text{in}} + \text{St}_{y,s,i,t}^{\text{out}} \quad \forall y \in Y, s \in S, t \in T \tag{3f}
\]

\[
\text{St}_{y,s,i,t}^{\text{size}} = \text{St}_{y,s,i,t}^{\text{size}} - 1 + \text{St}_{y,s,i,t}^{\text{in}} - \text{St}_{y,s,i,t}^{\text{out}} \quad \forall y \in Y, s \in S, i \in I^{\text{st}}, t \in T \tag{3g}
\]

\[
\text{Capa}_{y,s} = \text{Capa}_{y,s}^{(k)} + \lambda^{(k)}_{y,s} \quad \forall i \in I \tag{3h}
\]
Algorithm 1: Parallelized multi-cut benders, based on Conejo et al. (2006)

1. choose convergence tolerance $\epsilon$ and deletion threshold $\eta$
2. set $k = 1$ and $\xi_{up} \rightarrow \infty$
3. while $1 - \frac{\xi_{low}}{\xi_{up}} < \epsilon$ do
   4. solve MP get $\text{Capa}^{(k)}_{y,i}$ and $U^{(k)}$
   5. set $\xi_{low} \leftarrow \text{obj}(\text{MP})$
   6. do in parallel
      7. for $y \in Y, s \in S$ do
         8. solve $\text{SP}_{y,s}$ with $\text{Capa}^{(k)}_{y,i}$
         9. get $V^{(k)}_{y,s}$ and $\lambda^{(k)}_{y,i}$
        10. add $\Omega^{(k)}_{k,y,s}$ to MP
        11. set $\delta^{(k)}_{k,y,s} \leftarrow k$
      end for
    12. for $l \in \{1, \ldots, k-1\}, y \in Y, s \in S$ do
       13. if $\Omega^{(k)}_{l,y,s}$ is binding then
          14. set $\delta^{(k)}_{l,y,s} \leftarrow k$
       15. else if $k - \delta^{(k)}_{l,y,s} > \eta$ then
          16. delete $\Omega^{(k)}_{l,y,s}$ from MP
       end for
    end for
  18. if $\xi_{up} > U^{(k)} + \sum_{y \in Y, s \in S} p_s \cdot V^{(k)}_{y,s}$ then
     19. set $\xi_{up} \leftarrow U^{(k)} + \sum_{y \in Y, s \in S} p_s \cdot V^{(k)}_{y,s}$
  21. end if
22. set $k \leftarrow k + 1$
23. end while

Next, the Benders cuts $\Omega_{k,y,s}$ defined by Eq. 4 are computed and added to the MP. Each cut improves the estimator $\alpha_{y,s}$ based on the the exact result of the SP $V^{(k)}_{y,s}$ and its gradient $\lambda^{(k)}_{y,i}$ at the point $\text{Capa}^{(k)}_{y,i}$. In other words, by adding hyperplanes restricting the estimator, BD performs a piece-wise linear approximation of the unknown but convex function that assigns capacities to SP-costs (Conejo et al., 2006). In the literature, cuts of this form are termed optimality cuts, as opposed to feasibility cuts that are added when the SP was infeasible, which cannot occur in our case.

$$\Omega_{k,y,s} : \alpha_{y,s} \geq U^{(k)}_{y,s} + \sum_{i \in I} \lambda^{(k)}_{y,i} \cdot (\text{Capa}^{(k)}_{y,i} - \text{Capa}^{(k)}_{y,i}) \quad (4)$$

Afterwards, the value $\delta^{(k)}_{k,y,s}$ tracking in which iteration a cut was created or binding the last time is initialized based on the number of the current iteration $k$. Parallel to solving the SPs, line 13 to 18 of the algorithm loop over all previous cuts updating $\delta^{(k)}_{k,y,s}$ and deleting cuts that were not created or binding within the last $\eta$ iterations.

Once all SPs are solved, the total costs at $\text{Capa}^{(k)}_{y,i}$ are computed by summing expansion costs $U$ and accurate operational costs $V^{(k)}_{y,s}$. If these undercut the current $\xi_{up}$, $\text{Capa}^{(k)}_{y,i}$ is a new best solution and $\xi_{up}$ is adjusted accordingly. Finally, the iteration counter increases by one and the next iteration starts, if the optimality gap does not exceed the predefined tolerance yet.

The standard Benders implementation described in this section deploys all pre-existing refinements to the BD that we found sensible for the described two-stage planning problem. To solve the SPs faster, we use
the Barrier algorithm without crossover. Reducing the scenario number in early iterations is not sensible, because SPs are solved in parallel. We tested an approach for large SPs introduced in Mazzi, Grothey, McKinnon, and Sugishita (2021) that approximates SP solutions using a fitted saddle function, but the approach did not yield sufficiently accurate results in our case. We did not deploy advanced cut generation, because it requires solving the dual SP in addition to the primal SP, which is not beneficial for large SPs. Furthermore, we did not apply any further decomposition, for example Dantzig–Wolfe to address the complicating storage constraints in Eq. 3h of the SPs, or dual dynamic programming to solve the expansion problem with consecutive years sequentially (Sepúlveda, 2020). Our results below show that for the investigated continuous problem, decomposition is generally inferior to off-the-shelf solvers unless it enables distributed computing, which is already achieved by BD. Trust-region approaches are discussed in the subsequent section.

To be robust, the outlined algorithm is subject to several practical refinements. All capacity variables have an upper limit which is far beyond any reasonable value to pre-limit the solution space of the MP. To improve numerical properties, all equations are scaled according to the methodology outlined in (Göke, 2021a). If scaling does not suffice, small values of \( \text{Capa}^{(k)}_{y,i} \) in the SPs are rounded off to zero. The same applies to \( \lambda^{(k)}_{y,i} \) when adding cuts to the MP. For storage, we enforce an upper and lower bound on the ratio between storage and energy capacity that reflects technical restrictions and limits the solution space.

2.3. Dynamic \( l_2 \)-norm trust-region

Standard BD often suffers from slow convergence and solutions oscillate between the edges of the feasible region. In our case, this problem is particular pronounced, because the feasible region is hardly restricted and completely continuous. As a result, the algorithm moves very slowly from the edges to the optimal solution, like getting to the center of a sphere by making thin cuts from alternating sides.

The stability problems of BD are well acknowledged and addressed by various trust-region approaches. The key idea of these approaches is to avoid oscillation by restricting the feasible region artificially. There are two ways to achieve this by adapting the MP:

1. **Add a restricting constraint.** For binary problems, the \( l_0 \)-norm or Hamming distance can restrict the MP to a radius around a reference solution (Santoso et al., 2005). Continuous problems can use the \( l_\infty \)-norm that does not create a radius but limits the maximum step-size across all variables compared to a reference (Linderoth & Wright, 2003). In both cases, the radius or maximum step-size can be refined during iteration.

2. **Extend the objective with a penalty term.** To penalize deviation from the reference, the approach extends the objective with the \( l_2 \)-norm, or Euclidean distance, creating a quadratic problem (Ruszczyński, 2003).

The method we introduce combines both existing trust-region approaches. We add a constraint restricting the MP to a radius around a reference solution but use the Euclidean instead of the Hamming distance since the MP is continuous and not binary in our case. The constraint is enforced on the expansion variable and the radius set relative to the \( l^1 \)-norm of expansion variables for the current best solution \( \text{Exp}^{up} \). Eqs. 5a and 5b provide the formulation of the trust-region, using the \( l^p \)-norm and explicit notation, respectively. We use a relative radius based on the \( l^1 \)-norm, because it is more robust to absolute changes in capacity and across applications.

\[
\sum_{y \in Y, i \in I} (\text{Exp}_{y,i} - \text{Exp}^{up}_{y,i})^2 \leq (\phi \cdot \sum_{y \in Y, i \in I} \text{Exp}^{up}_{y,i})^2 \quad \forall y \in Y, s \in S
\]
While iterating the trust-region is dynamically adapted based on two rules: First, if the current best solution improves and the upper bound $\zeta_{up}$ decreases, the trust-region is re-centered, and its radius adjusted accordingly. Second, if the trust-region is not a binding constraint for the MP, its relative radius $\phi$ is halved. The approach is not at risk of converging to a local optimum within the trust-region because the problem is linear. For the same reason, the trust-region is only non-binding if the optimum is already within it and reducing the radius is sensible.

In the following, we illustrate these rules based on four consecutive iterations for a stylized example problem in Fig. 4. It includes three unbounded expansion variables, restricted by a spherical trust-region according to Eq. 5b. The optimal solution is indicated in green, the current best in orange, and the solution at the current iteration $Exp^{(k)}$ in yellow or red, depending on whether it improves the current best or not. The first iteration in (a) results in an objective above the current best, but the trust-region is binding and remains unchanged. Due to the added cuts, the second iteration (b) improves the current best and the trust-region is consequently re-centered for the third iteration. Therefore, its radius is also adapted based on the $l^1$-norm of the new solution. The third iteration in (c) does not improve the current best, but the trust-region is not binding anymore, and its radius halved accordingly. The fourth iteration with a smaller trust-region in (d) does not improve the solution.

**Figure 4:** Adaption of trust-region during iteration

As an added benefit, trust-regions enable BD to benefit from initial heuristic solutions. For standard BD,
starting solutions do not improve convergence, because lacking cuts approximating the SPs, the algorithm will not identify the quality of a solution and move to the edges of the solution space regardless. But enforcing a trust-region based on heuristics improves the quality of cuts, especially in early iterations, since BD is forced into the neighborhood of the optimum. The following sections will investigate, how initializing the quadratic trust-region with the solution of the deterministic problem affects convergence.

Alg. 2 extends the BD algorithm introduced in the previous section according to the dynamic trust-region approach.

If the algorithm uses initialization, it first solves the closed problem (CP) for the scenario with the highest probability $p_s$. Afterwards, all SPs are solved in parallel for $Capa_{y,i}^{(0)}$ to set $\zeta_{ap}$ and the corresponding Benders cuts are added to the MP. Without initialization, the steps prior to iteration do not change.

During iteration, the algorithm now solves the quadratic master-problem (QMP) including the trust-region to determine $Capa_{y,i}^{(k)}$. Parallel to solving the SPs for $Capa_{y,i}^{(k)}$, the regular MP is solved as well to determine its objective value and expansion costs. The MP’s objective then sets the lower bound $\varepsilon_{low}$ since the trust-region distorts the QMP’s objective. The expansion costs of the MP and QMP, $U_{std}^{(k)}$ and $U_{qua}^{(k)}$ respectively, determine, if the trust-region is binding and the relative trust-region radius $\phi$ is halved in line 42. In terms of coding, this approach is more convenient than explicitly computing the slack of the actual constraint. The trust-region is only halved, if the current iteration did not change the current best solution $Exp_{y,i}^{ap}$. If the iteration did improve the current best, the trust-region is updated in line 39 not only moving its center but, considering the left-hand side of Eq. 5b, also adapting its radius. As a result, the absolute radius of the trust-region can decrease making the updated trust-region binding again.

Compared to regularized trust-region methods that add a quadratic term to the objective, our approach does not interfere with convergence. If the current best solution is poor, moving towards better solutions is not slowed down by a penalty. If the current best is close to the optimum, the approach is not at risk of converging to a local minimum because the penalty exceeds the improvement of the objective. Thanks to adaption of the radius, the approach still supports convergence in late iterations. At the same time, the complexity added by a quadratic constraint is small, if the MP is easy, especially due to recent improvements of commercial solvers. Solving the regular MP as well does not affect computation time, because it is fully parallelized.
Algorithm 2: Parallelized multi-cut benders with dynamic quadratic trust-region

1. chose convergence tolerance $\epsilon$ and deletion threshold $\eta$
2. chose initial relative radius $\phi$
3. if initialize trust-region true then
   4. solve CP with $S' := \{ s \in S \mid p_s = \max(p_s) \}$ and get $Exy_{y,i}^{(0)}$
   5. do in parallel
      6. for $y \in Y, s \in S$ do
         7. solve $SP_{y,s}$ with $Capa_{y,i}^{(0)}$
         8. get $V_{y,s}^{(0)}$ and $\lambda_{y,i}^{(0)}$
         9. add $\Omega_{0,y,s}$ to QMP and MP
         10. set $\delta_{0,y,s} \leftarrow 0$
      end for
   11. end
   12. set $\zeta_{up} \leftarrow U^{(0)} + \sum_{y \in Y, s \in S} p_s \cdot V_{y,s}^{(0)}$
   13. set $Exp_{y,i}^{(0)} \leftarrow Exp_{y,i}^{(0)}$ and update trust-region of QMP
14. else
   15. set $\zeta_{up} \rightarrow \infty$
16. end if
17. set $k \leftarrow 1$
18. while $1 - \frac{\zeta_{low}}{\zeta_{up}} < \epsilon$ do
19.   solve QMP and get $Capa_{y,i}^{(k)}, Exp_{y,i}^{(k)}$ and $U_{qua}^{(k)}$
20.   do in parallel
21.      for $y \in Y, s \in S$ do
22.         solve $SP_{y,s}$ with $Capa_{y,i}^{(k)}$
23.         get $V_{y,s}^{(k)}$ and $\lambda_{y,i}^{(k)}$
24.         add $\Omega_{k,y,s}$ to QMP and MP
25.         set $\delta_{k,y,s} \leftarrow k$
26.      end for
27.      for $l \in \{1, \ldots, k-1\}, y \in Y, s \in S$ do
28.         if $\Omega_{l,y,s}$ is binding then
29.             set $\delta_{l,y,s} \leftarrow k$
30.         else if $k - \delta_{l,y,s} > \eta$ then
31.             delete $\Omega_{l,y,s}$ from MP
32.         end for
33.      solve MP and get $U_{std}^{(k)}$
34.      set $\zeta_{low} \leftarrow obj(MP)$
35.   end
36.   if $\zeta_{up} > U_{std}^{(k)} + \sum_{y \in Y, s \in S} p_s \cdot V_{y,s}^{(k)}$ then
37.      set $\zeta_{up} \leftarrow U_{std}^{(k)} + \sum_{y \in Y, s \in S} p_s \cdot V_{y,s}^{(k)}$
38.      set $Exp_{y,i}^{up} \leftarrow Exp_{y,i}^{(k)}$ and update trust-region of QMP
39.   else
40.      if $U_{std}^{(k)} = U_{qua}^{(k)}$ then
41.         set $\phi \leftarrow 0.5 \cdot \phi$ and update trust-region of QMP
42.      end if
43.   end if
44.   set $k \leftarrow k + 1$
45. end while
3. Case study

This section introduces the case-study used to test and benchmark the algorithms introduced previously. The ambition of the case-study is not to achieve highly accurate system planning, but to provide a plausible problem covering all critical elements of renewable energy systems. All files to re-produce the case-study are placed in a public repository linked in the Supplementary material section.

3.1. Applied planning model

The applied planning model includes two consecutive years, 2030 and 2040, and focuses on the power sector. To cover a diverse range of system setups, the planned system is not subject to any emission constraints in 2030 but must fully decarbonize until 2040. Spatially the model covers four distinct regions: France, Belgium, the Netherlands, and Germany.

Figure 5 introduces all considered technologies, depicted as gray circles, and their interaction with energy carriers, depicted as colored squares. Exogenous demand in the model is limited to the carrier electricity which is modeled at an hourly resolution for the entire year. Hydrogen uses a daily resolution; oil and gas a yearly resolution.

In the graph, entering edges of technologies refer to their input carriers; outgoing edges relate to outputs. For example, the generation technology electrolysis uses electricity as an input to generate hydrogen. Storage technologies, like pumped storage, have an entering and an outgoing edge to represent charging and discharging. Beyond pumped storage, the model includes batteries for short-term and hydrogen tanks for long-term energy storage. Including these technologies is critical due to their importance for decarbonized systems and their impact on the structure of the planning problem.

Capacities for wind and PV are subject to an upper limit that restricts expansion. Pumped storage, run-of-river, and hydro reservoirs have capacities fixed to today’s levels. Furthermore, hydro reservoirs are modelled as storages with an exogenous inflow instead of flexible charging. Beyond the listed technologies, the model also decides on expansion and operation of transmission to exchange electricity and hydrogen between regions.
3.2. Scenarios for climatic years

To compare the introduced variations of BD, we solve the model as a two-stage planning problem with multiple scenarios. Each scenario represents a climatic year with different patterns and total levels of electricity demand and capacity factors for wind and PV. For this purpose, we build on corresponding data for 39 climatic years from 1980 to 2018 published in H. Bloomfield, Brayshaw, and Charlton-Perez (2020), based on re-analysis data from the NASA’s MERRA-2 dataset.

Stochastic programming often reduces $n$ scenarios of historical data, in our case 39 climatic years, to a smaller number $m < n$ that captures the empirical distribution but improves computational tractability (Dupačová, J. and Grüwe-Kuska, N. and Römisch, W., 2003; Rujeerapaiboon, Schindler, Kuhn, & Wiesemann, 2022). Scenario reduction methods determine a set of representative scenarios $S$ and a set of corresponding weights $\{p_s : s \in S, \sum_{s \in S} p_s = 1, p_s \geq 0\}$, such that the weighted scenario average of second stage costs approximates the sample average over all $n$ scenarios. There is a variety of reduction methods for this purpose ranging from naïve sampling to moment matching or clustering methods (Kaut, 2021). The choice of scenario reduction method and the optimal number of scenarios $m$ for adequate planning of renewable systems is an open question which is beyond the scope of this paper. For a meaningful comparison of solution methods, as conducted in Section 4 below, it is sufficient to ensure the $m$ selected scenarios are not extremely similar and bias the algorithmic performance as a result.

In this paper, we used a problem-dependent $k$-medioid algorithm, as proposed by Bertsimas and Mundru (2022), for selecting $m$ representative scenarios and corresponding probability weights. The method is advantageous for large-scale planning problems since it reduces the high-dimensional clustering problem including hourly data on demand and capacity factors across multiple countries to a much smaller clustering problem in $\mathbb{R}_+$, the space of system costs. To perform the clustering, we calculate a matrix of symmetric distances between scenarios, or climatic years, where the $s$-th row and $s^*$-th column is computed according to Eq. 6:

$$d_{s,s^*} = \frac{1}{2} [Z_{s,s^*} - Z_{s,s} + Z_{s^*,s} - Z_{s^*,s^*}] \tag{6}$$

where $Z_{s,s^*}$ refers to total system costs when the deterministic problem for scenario $s$ is solved with capacities fixed to results of the deterministic problem for scenario $s^*$. For instance, $Z_{1980,2008}$ corresponds to total system costs, including infeasibility costs, when solving the deterministic model for the climatic year 1980 with capacities originally computed for 2008. The clustering algorithm selects $m$ representative scenarios and assigns each of the $n - m$ scenarios remaining in the sample to one of the representative scenarios, such that the sum of distances between representative scenarios and assigned scenarios as defined above, is minimized.

Fig. 6 illustrates the results of the clustering algorithm for $m = 6$. Each color represents a group and the node of the representative scenario for this group is equipped with its weight $p_s$, that reflects the relative group size. The color and thickness of connecting lines between the nodes indicates similarity, the inverse of the distance $d_{s,s^*}$ as defined above.

---

2The sample average in turn is assumed to converge asymptotically to the expected cost of the second stage, provided a suitable law of large numbers holds.
4. Results

This section first compares the introduced variations of BD and then BD against closed optimization. The comparison is based on solving the case-study while varying the number of stochastic scenarios from 2 to 16. The MP and each SP run on independent nodes of a computing cluster, each with four cores and 16 GB of memory. Accordingly, the number of nodes amounts to one for the MP plus twice the scenario number for the SPs, since the case-study covers two consecutive years, 2030 and 2040.

For solving problems, each node deploys the Barrier implementation of Gurobi without Crossover, the NumericFocus parameter set to zero, and the convergence tolerance to 1e-4 for the SPs. For BD, we delete unused cuts after 20 iterations, set the initial relative radius of the trust-region to 0.01, and choose a convergence tolerance of 0.2%. At smaller tolerances the algorithm sometimes fails to converge due to rounding (see end of section 2.2). All deployed variations of BD are implemented in the version of the AnyMOD.jl modelling framework linked in the Supplementary material.

4.1. Comparison of Benders versions

First, we compare the standard BD without trust-region described in section 2.2 against our refined trust-region method with and without initialization based on a heuristic solution. For all three variations and two, four, or six stochastic scenarios, Fig. 7 plots the runtime against the optimality gap after each iteration. Consequently, points for BD with initialization are offset to the right due to the run-time spent on the heuristic solution previous to iteration.

Overall Fig. 7 demonstrates the substantial speed-up the trust-region approach achieves over the standard algorithm decreasing the solution time by a factor of 4 to 6 depending on the number of scenarios. The acceleration results from greatly reducing the number of iterations, which outweighs the increase in time per iteration due to the added quadratic constraint. With two scenarios for instance, average time per iteration increases from 22 to 30 seconds when using the trust-region method, but at the same time the number of iterations drops from 1438 to 181 when initializing the trust-region and to 279 when not. Initializing the trust-region only has a minor and inconclusive impact on the total solution time. Increasing the number of

---

3Core models can differ between runs and nodes. To ensure hardware differences do not bias results, conspicuous performance outliers were re-run always reproducing the initial results.
Benders decomposition with quadratic trust-region scenarios does not have a consistent effect as well, because all algorithms are fully parallelized.

Furthermore, adding storage technologies has a much greater effect on iterations and solution time than adding generation technologies. In contrast to generation, storage is not constrained by a single capacity variable, but multiple interlinked capacities and their interaction is difficult to capture for BD. For instance, if only the capacity constraints on storage output are binding in a SP, the shadow variable $\lambda^{(k)}_{y,i}$ is large for power output, but zero for energy capacity. Consequently, the algorithm will solely expand output capacity in the following iteration and not consider how energy capacity constrains operation as well.

4.2. Comparison against closed optimization

This section compares closed optimization without decomposition against BD with quadratic trust-regions increasing the number of scenarios up to 16.

The significance of this comparison has its limits because closed optimization and BD utilize computational resources differently. Closed optimization with barrier is restricted to a single node, but BD utilizes multiple nodes and distributed computing. In our case, closed optimization runs on a single node with 8 cores and 64 GB memory, the most powerful node available to us, while BD utilizes up to 33 nodes with 4 cores and 16 GB memory each. Considering the benchmark is not universally valid and depends on the computational resources at hand, we avoid juxtaposing solution times and rather focus on the scaling behaviour.

Fig. 8 shows the run-time to generate and solve the problem for closed optimization using between 2 and 16 scenarios. Generation and solve time both increase exponentially with the number of scenarios, but solve time has an outlier at 12 scenarios. The outlier was reproduced several times and likely results from the mechanics of the barrier algorithm.

On the other hand, Fig. 9 shows the run-time for the same number of scenarios using the introduced trust-region approach with and without initialization. Thanks to decomposition and parallelization, neither run-time of model generation nor solution increase with the number of scenarios. Time for model generation itself is virtually constant and only computing the heuristic solution when initializing the trust-region on the right side of the graph adds some fluctuations. Running BD and solving the model is subject to greater fluctuations, but BD generally outperforms closed optimization, if the number of scenarios exceeds six.

In terms of solution quality, BD performs favorable as well. Since closed optimization uses barrier without crossover, none of the two methods computes an exact minimum, but with BD, objective values are on average 0.44% smaller. For 12 scenarios, the objective value with BD is even 2.44% smaller than the result of closed optimization, which ties in with the outlier in solve time observed earlier.
Finally, the results show that on average BD with initialization is slightly faster than without. Speed-up of the iteration tends to exceed the added run-time of first solving only with the most probable scenario to obtain a starting solution.

5. Conclusion

This paper applied BD to two-stage stochastic problems for energy planning with multiple climatic years, a key problem for the design of renewable energy systems. First, we implemented BD with existing enhancements suited for the characteristics of the problem, a simple continuous MP and few but large SPs. Next, we developed a novel trust-region method using a quadratic constraint that is continuously adapted to further speed-up the algorithm. In a quantitative case-study our method accelerates BD by a factor of four to six slightly increasing solve time of the MP, but greatly reducing the number of iterations. With the computational resources at our disposal, our refined BD outperforms closed optimization if planning covers more than six climatic years, because run-time does not increase with the number of scenarios thanks to distributed computing. Furthermore, results show that the quadratic trust-region approach benefits from a heuristic starting solution but does not depend on it to be performative.

Although the literature does not offer any further enhancements of BD that are directly applicable to planning renewable energy systems with multiple climatic years, there are still several promising approaches for further improvement. To present them, we again follow the categorization of enhancements introduced in Rahmaniani et al. (2017):

- **Solution procedure**: Considering most of the run-time is spent on the SPs, solving approximations...
instead of the original SPs can further increase performance, especially in early iterations. Not solving SPs to optimality in the current algorithm already exploits that valid cuts do not require exact solutions. Conceivable approximation methods include surrogate models based on machine learning, geometric interpolation, or using a reduced temporal resolution (Göke & Kendziorski, 2021; Mazzi et al., 2021; Neumann & Brown, 2021). Approximations face two challenges though: First, the high dimensionality of inputs to the SP that corresponds to the number of complicating capacity variables. Second, valid cuts must build on SP solutions that are not necessarily optimal but feasible. This implies approximations cannot underestimate the objective of a SP.

• **Solution generation**: Although heuristic methods are insufficient for adequate planning, they can greatly narrow the solution space of the MP to improve convergence. This approach also offers synergies with the selection of representative climatic years for the planning problem in the first place. For instance, solving the deterministic problem for specific climatic years, as performed in section 3.2, is not only useful for clustering climatic years, but can also derive lower and upper bounds for capacity variables or system costs.

• **Cut generation**: The disproportional impact storage technologies have on the run-time that was discussed at the end of section 4.1 suggests the algorithm would benefit from advanced cut generation. However, the size of the SPs still poses an obstacle for this approach. Therefore, refined approaches that only solve a modified but not the original SP are most promising (Papadakos, 2008; Sherali & Lunday, 2011).

Apart from further enhancements, various practical applications benefit from the introduced BD algorithm. First and foremost, it enables more robust planning of renewable macro-energy systems by including a large number of climatic years, which was the original motivation for this work. How many climatic years to include and how to select these years for adequate system planning is an open question for future research. In addition, future research should extend the optimization under uncertainty to the operational stage in the SPs. Currently, uncertainty is limited to expansion in the MP and operation in each scenario assumes perfect foresight, but stochastic dual dynamic programming could capture uncertainty at the operational stage as well (Papavasiliou, Mou, Cambier, & Scieur, 2018).

Finally, our trust-region method benefits energy planning models that are deterministic and limited to a single climatic year, but capture learning effects using piecewise-linear cost curves (Zeyen, Victoria, & Brown, 2022). In this case, linearization adds mixed-integer variables to the expansion problem and as result even standard BD outperforms closed optimization (Felling, Levers, & Fortenbacher, 2022). Generally, mixed-integer variables in the MP will increase the advantage over closed optimization and could even speed-up BD when reducing the solution space; for instance, when modelling how expansion of large-scale technologies, like nuclear plants, is rather a discrete than a continuous decision.

6. Acknowledgments

The research leading to these results has received funding from the German Federal Ministry for Economic Affairs and Energy via the project "MODEZEEN" (grant number FKZ 03EI1019D). A special thanks goes to all Julia developers. A special thanks goes to all Julia developers.

Supplementary material

The data and execution scripts for the case-study are available here: https://github.com/leonardgoeke/EuSysMod/releases/tag/bendersDecompositionPaper. The underlying AnyMOD.jl version imple-
menting the refined BD algorithm is released here: https://github.com/leonardgoeke/AnyMOD.jl/releases/tag/bendersDecompositionPaper.

References

Backe, S., Skar, C., del Granado, P. C., Turgut, O., & Tomaszgard, A. (2022). Empire: An open-source model based on multi-horizon programming for energy transition analyses. SoftwareX, 17, 100877. doi: 10.1016/j.softx.2021.100877

Benders, J. (1962). Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik, 4, 238-252.

Bertsimas, D., & Mundru, N. (2022). Optimization-based scenario reduction for data-driven two-stage stochastic optimization. Operations Research, doi: 10.1287/opre.2022.22625.

Bloomfield, H., Brayshaw, D., & Charlton-Perez, A. (2020). Merra2 derived time series of european country-aggregate electricity demand, wind power generation and solar power generation. doi: 10.17864/1947.239

Bloomfield, H. C., Brayshaw, D. J., Shaffrey, L. C., Coker, P. J., & Thornton, H. E. (2016). Quantifying the increasing sensitivity of power systems to climate variability. Environmental Research Letters, 11(12), 124025. doi: 10.1088/1748-9326/11/12/124025

Brandenberg, R., & Stursberg, P. (2021). Refined cut selection for benders decomposition: applied to network capacity expansion problems. Mathematical Methods of Operations Research, 94(3), 383-412. doi: 10.1007/s00186-021-00756-

Collins, S., Deane, P., Ó Gallachóir, B., Pfenninger, S., & Staffell, I. (2018). Impacts of inter-annual wind and solar variations on the European power system. Joule, 2(10), 2076-2090. doi: 10.1016/j.joule.2018.06.020

Conejo, A., Castillo, E., Minguez, R., & Garcia-Bertrand, R. (2006). Decomposition techniques in mathematical programming: Engineering and science applications. Springer Berlin Heidelberg.

Creutzig, F., Agoston, P., Goldschmidt, J. C., Luderer, G., Nemet, G. F., & Pietzcker, R. C. (2017). The underestimated potential of solar energy to mitigate climate change. Nature Energy, 2, 17140.

Dupačová, J. and Grówe-Kuska, N. and Römisch, W. (2003). Scenario reduction in stochastic programming - an approach using probability metrics. Mathematical Programming, 95, 493–511. doi: 10.1007/s10107-002-0331-0

Felling, T., Levers, O., & Fortenbacher, P. (2022). Multi-horizon planning of multi-energy systems. Electric Power Systems Research, 212, 108509. doi: 10.1016/j.epsr.2022.108509

Göke, L. (2021a). Anymod.jl: A julia package for creating energy system models. SoftwareX, 16, 100871. doi: 10.1016/j.softx.2021.100871

Göke, L. (2021b). A graph-based formulation for modeling macro-energy systems. Applied Energy, 301, 117377. doi: 10.1016/j.apenergy.2021.117377

Göke, L., & Kendziorksi, M. (2021). The adequacy of time-series reduction for renewable energy systems. Energy, 238, 121701. doi: 10.1016/j.energy.2021.121701

Göke, L., Weibezahn, J., & von Hirschhausen, C. (2021a). Fictional expectations in energy scenarios and implications for planning macro-energy systems. Working Paper. Retrieved from https://arxiv.org/abs/2112.04821

Hilbers, A. P., Brayshaw, D. J., & Gandy, A. (2019). Importance subsampling: improving power system planning under climate-based uncertainty. Applied Energy, 251, 113114. doi: 10.1016/j.apenergy.2019.04.110

Kaut, M. (2021). Scenario generation by selection from historical data. Computational Management Science, 18, 411-429. doi: 10.1007/s10287-021-00399-4

Linderoth, J., & Wright, S. (2003). Decomposition algorithms for stochastic programming on a computational grid. Computational Optimization and Applications, 24(2), 207–250. doi: 10.1023/A:1021858008222

Lohmann, T., & Rebennack, S. (2017). Tailored benders decomposition for a long-term power expansion model with short-term demand response. Management Science, 63(6), 2027-2048. doi: 10.1287/mnsc.2015.2420

Lombardi, F., Rocco, M. V., Belussi, L., Danza, L., Magni, C., & Colombo, E. (2022). Weather-induced variability of country-scale space heating demand under different refurbishment scenarios for residential buildings. Energy, 239, 122152. doi: 10.1016/j.energy.2021.122152

Magnanti, T. L., & Wong, R. T. (1981). Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. Operations Research, 29(3), 464-484. doi: 10.1287/opre.29.3.464

Mazzi, N., Grothey, A., McKinnon, K., & Sugishita, N. (2021). Benders decomposition with adaptive oracles for large scale optimization. Mathematical Programming Computation, 13(1), 683–703. doi: 10.1007/s12532-020-00197-0

Neumann, F., & Brown, T. (2021). Broad ranges of investment configurations for renewable power systems, robust to cost uncertainty and near-optimality. arXiv. doi: 10.48550/ARXIV.2111.14443

Ohlendorf, N., & Schill, W.-P. (2020). Frequency and duration of low-wind-power events in Germany. Environmental Research Letters, 15(8), 084045. doi: 10.1088/1748-9326/ab91e9

Papadakos, N. (2008). Practical enhancements to the magnanti–wong method. Operations Research Letters, 36(4), 444-449. doi: 10.1016/j.orl.2008.01.005

Papavasiliou, A., Mou, Y., Cambier, L., & Scieur, D. (2018). Application of stochastic dual dynamic programming to the real-time dispatch of storage under renewable supply uncertainty. IEEE Transactions on Sustainable Energy, 9(2), 547-558. doi: 10.1109/TSTE.2017.2748463

Pfenninger, S. (2017). Dealing with multiple decades of hourly wind and pv time series in energy models: A comparison of methods to reduce time resolution and the planning implications of inter-annual variability. Applied Energy, 197, 1-13. doi: 10.1016/j.apenergy.2017.03.051

Pfenninger, S., & Staffell, I. (2016). Long-term patterns of european pv output using 30 years of validated hourly reanalysis and satellite data. Energy, 114, 1251-1265. doi: 10.1016/j.energy.2016.08.060

Rahmaniani, R., Crainic, T. G., Gendreau, M., & Rei, W. (2017). The benders decomposition algorithm: A literature review. European Journal of Operational Research, 259(3), 801-817. doi: 10.1016/j.ejor.2016.12.005

Ritchie, H., & Roser, M. (2020). Energy. Retrieved from https://ourworldindata.org/energy
Benders decomposition with quadratic trust-region

Ruhnau, O., & Qvist, S. (2022). Storage requirements in a 100% renewable electricity system: extreme events and inter-annual variability. *Environmental Research Letters, 17*(4), 044018. doi: 10.1088/1748-9326/ac4dc8

Rujeerapaiboon, N., Schindler, K., Kuhn, D., & Wiesemann, W. (2022). Scenario reduction revisited: fundamental limits and guarantees. *Mathematical Programming, 191*, 207–242. doi: 10.1007/s10107-018-1269-1

Ruszczyński, A. (2003). Decomposition methods. In *Stochastic Programming* (Vol. 10, p. 141-211). doi: 10.1016/S0927-0507(03)10003-5

Santoso, T., Ahmed, S., Goetschalckx, M., & Shapiro, A. (2005). A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research, 167*(1), 96-115. doi: 10.1016/j.ejor.2004.01.046

Sepulveda, N. (2020). *Decarbonization of power systems, multi-stage decision-making with policy and technology uncertainty* (Doctoral dissertation, Massachusetts Institute of Technology, Department of Nuclear Science and Engineering.) Retrieved from https://hdl.handle.net/1721.1/127311

Sherali, H., & Lunday, B. (2011). On generating maximal nondominated benders cuts. *Annals of Operations Research, 210*, 57–72. doi: 10.1007/s10479-011-0883-6

Skar, C., Doorman, G., & Tomasgard, A. (2014). Large-scale power system planning using enhanced benders decomposition. In *2014 power systems computation conference* (p. 1-7). doi: 10.1109/PSCC.2014.7038297

Teichgraeber, H., Küpper, L. E., & Brandt, A. (2021). Designing reliable future energy systems by iteratively including extreme periods in time-series aggregation. *Applied Energy, 304*, 117696. doi: 10.1016/j.apenergy.2021.117696

van Ackooij, W., Lebbe, N., & Malick, J. (2017). Regularized decomposition of large scale block-structured robust optimization problems. *Computational Management Science, 14*(3), 393-421. doi: 10.1007/s10287-017-0281-x

Van Slyke, R. M., & Wets, R. (1969). L-shaped linear programs with applications to optimal control and stochastic programming. *SIAM Journal on Applied Mathematics, 17*(4), 638-663. doi: 10.1137/0117061

Zakeri, G., Philpott, A. B., & Ryan, D. M. (2000). Inexact cuts in benders decomposition. *SIAM Journal on Optimization, 10*(3), 643-657. doi: 10.1137/S1052623497318700

Zeyen, E., Victoria, M., & Brown, T. (2022). *Endogenous learning for green hydrogen in a sector-coupled energy model for europe*. arXiv. doi: 10.48550/ARXIV.2205.11901