Electronic Supplement: Decomposition Techniques for Model-Based Bid Selection in Renewable Auctions

Alexandros Visas, Daniel Ávila, Anthony Papavasiliou,
Georg Thomassen, and David Pozo

1 Introduction

This electronic supplement provides additional details and derivations for the main paper [1].

Section 2 complements Section III of the main manuscript by giving the details of the stabilized Benders decomposition algorithm. It describes the fixed parameter level set stabilization algorithm, the dynamic level parametrization variant, and the cut consolidation, cut sparsification, and valid inequalities used to strengthen and control the size of the LP and QP master problems[1].

Section 3 provides additional results that complement the CWE case study of Section V of the main manuscript. It first examines the effect of temporal segmentation, then compares algorithmic variants that motivate the algorithmic choice used in the full-EU experiments, and finally reports sensitivity analyses.

Section 4 provides additional results that complement the full-EU results of Section V of the main manuscript.

2 Decomposition algorithm details

This section presents the algorithmic components that underlie the proposed stabilized Benders decomposition scheme. Section 2.1 first presents the fixed parameter level set stabilization algorithm, summarized in Algorithm S1. Section 2.2 then introduces the dynamic parametrization of the level set stabilization algorithm through Routine S2, and gives the corresponding dynamic level set variant in Algorithm S3. Section 2.3 describes the cut consolidation and cut sparsification procedures used to control the growth of the master problems. Section 2.4 presents the valid inequalities used to strengthen the master problems and then combines the preceding components into the proposed decomposition algorithm, reported as Algorithm S4, which corresponds to Algorithm 1 of the main paper.

[1] In this supplement, the terms LP master and lower bound master are used interchangeably. Likewise, the terms QP master and level set master are used interchangeably.

Algorithms S1, S3, and S4 all use the same initialization strategy. When the deterministic wait-and-see problems are tractable, we solve one such problem for each scenario and define the initial stability center $\widehat{z}^0 = (\widehat{x}^0, \widehat{q}^0, \widehat{v}^0)$ as the average of the resulting x, q, and v decisions. For the full-EU instance, where the monolithic wait-and-see solves are too large, the regional deterministic initialization described in the main manuscript is used for the same purpose. This initialization gives the level set master a meaningful initial stability center before many Benders cuts have been generated. In our experiments, it reduced the number of iterations and the time to termination, suggesting that such initialization can be useful in practical applications that require solving extreme scale stochastic problems.

2.1 Fixed parameter level set stabilization

This subsection provides the details of the fixed parameter level set stabilization algorithm described in Section III-D of the main manuscript [1]. The two master formulations used throughout are the lower bound master problem (M) and the level set master problem (LS-M), both introduced there.

Classical Benders decomposition builds a cutting-plane approximation of the expected second stage operating cost. In the lower bound master problem, the variables $\theta_{k,\omega}$ approximate the segment-scenario operating costs and are constrained by the optimality cuts generated in previous iterations. The solution of (M) is used as the next trial point and the objective value of (M) gives the lower bound LB^j. This can be inefficient in early iterations, when the cutting-plane approximation is still weak, because consecutive trial points may jump between distant regions of the feasible set. This oscillatory behavior can lead to many iterations and slow practical convergence in large scale energy planning models.

Level set stabilization separates the computation of the lower bound from the selection of the next trial point. We use $z = (x, q, v)$ for the first stage decisions of the temporally decomposed master problem. Here, x is the investment vector, $q_{k,\omega}$ are the segment-scenario net renewable production budgets, and $v_{t_k,\omega}$ are the stored energy boundary variables.

For an evaluated trial point $\bar{z}^j = (\bar{x}^j, \bar{q}^j, \bar{v}^j)$, the upper bound is computed by adding the investment cost at $\bar{x}^j$ to the operating cost obtained by summing $\mathcal{V}_{k,\omega}(\bar{z}^j)$ over temporal segments and averaging over scenarios:

$$UB^j = \sum_{n \in N,\, g \in G^{\mathrm{INV}}(n)} IC_g\, \bar{x}^j_g + \mathbb{E}_\omega \left[\sum_{k \in \mathcal{K}} \mathcal{V}_{k,\omega}(\bar{z}^j) \right] \tag{1}$$

At iteration j, the algorithm first solves (M), and its objective value gives LB^j. The incumbent upper bound available before this iteration is $(UB^{j-1})^* = \min_{0 \le \nu < j} UB^\nu$. The evaluated trial point associated with this incumbent upper bound is the stability center $\widehat{z}$.

The fixed level parameter $\lambda \in (0, 1)$ determines where the level imposed in (LS-M) is placed between the current lower bound and the incumbent upper

bound. At iteration j, this level is computed by:

$$\ell^j(\lambda) = \lambda \cdot LB^j + (1-\lambda) \cdot (UB^{j-1})^* \tag{2}$$

Thus, larger values of λ move the level closer to the current lower bound LB^j, while smaller values move it closer to the incumbent upper bound $(UB^{j-1})^*$. In the fixed parameter scheme, $\lambda^j = \lambda$ for all iterations.

$(LS\text{-}M)$ then uses this level to select the trial point $\bar{z}^j$. It contains the investment constraints, the master constraint on the net renewable production budgets, the bounds on the stored energy boundary variables, and the accumulated optimality cuts, and adds the level constraint:

$$\sum_{n \in N,\, g \in G^{\mathrm{INV}}(n)} IC_g \, x_g + \mathbb{E}_\omega \left[\sum_{k \in \mathcal{K}} \theta_{k,\omega} \right] \leq \ell^j(\lambda) \tag{3}$$

The left hand side of (3) is the lower bound master objective. Among the points satisfying this constraint, $(LS\text{-}M)$ minimizes $\|z - \widehat{z}\|_2^2$. The trial point $\bar{z}^j = (\bar{x}^j, \bar{q}^j, \bar{v}^j)$ is therefore chosen to remain close to the stability center while satisfying the prescribed level for the current cutting-plane approximation.

After $\bar{z}^j$ has been obtained, the segment-scenario subproblems are evaluated at this trial point. The same subproblem solves are then used to generate the Benders optimality cuts. In particular, for each segment $k \in \mathcal{K}$ and scenario $\omega \in \Omega$, the cut $c^j_{k,\omega}$ is constructed from the dual variables associated with the subproblem constraints in which the first stage decisions $\bar{x}^j$, $\bar{q}^j$, and $\bar{v}^j$ appear. These cuts are added to both master problems and refine the cutting-plane approximation for later iterations. If UB^j improves the incumbent upper bound $(UB^{j-1})^*$, then the stability center is updated as $\widehat{z} \leftarrow \bar{z}^j$. Otherwise, it remains unchanged. Algorithm S1 summarizes these steps.

Algorithm S1 Benders Decomposition with fixed Parameter Level Set Stabilization

INPUT: λ, $j_{\max}$, ϵ_{tol}, and $\tau_{\max}$[2].
INITIALIZE: Compute $\widehat{z}^0$ by solving the deterministic wait-and-see problems and averaging the resulting first stage decisions. Set $\widehat{z} \leftarrow \widehat{z}^0$, initialize $(UB^0)^*$ with a sufficiently large value, set $j \leftarrow 1$ and $\epsilon^0 \leftarrow +\infty$.
while $j \leq j_{\max}$, $\epsilon^{j-1} > \epsilon_{\text{tol}}$, and elapsed time $< \tau_{\max}$ **do**
  Solve (M) and set LB^j equal to its objective value.
  Set $\lambda^j \leftarrow \lambda$ and compute $\ell^j(\lambda^j)$ from (2).
  Solve $(LS\text{-}M)$ and obtain $\bar{z}^j = (\bar{x}^j, \bar{q}^j, \bar{v}^j)$.
  for $k \in \mathcal{K}$, $\omega \in \Omega$ **do**
    Solve $\mathcal{V}_{k,\omega}(\bar{z}^j)$ and generate $c^j_{k,\omega}$.
  end for
  Compute UB^j from (1) and set $(UB^j)^* \leftarrow \min\{(UB^{j-1})^*, UB^j\}$.
  If $(UB^j)^* < (UB^{j-1})^*$, set $\widehat{z} \leftarrow \bar{z}^j$.
  Set $\epsilon^j \leftarrow ((UB^j)^* - LB^j)/LB^j$.
  Add $c^j_{k,\omega}$, $k \in \mathcal{K}$, $\omega \in \Omega$, to (M) and $(LS\text{-}M)$.
  Set $j \leftarrow j + 1$.
end while

2.2 Level set stabilization with dynamic parametrization

This subsection complements Section III-E of the main paper [1] by detailing how the level parameter is updated dynamically in the stabilized Benders decomposition algorithm. The fixed parameter scheme of Section 2.1 requires a single value of λ to be chosen before the run. This can be restrictive in our application, where the average wait-and-see initialization often provides a good incumbent upper bound early, while later progress is mainly obtained through improvements in the lower bound.

The dynamic parametrization routine presented in this section is motivated by [2]. However, we modify it to better reflect the bound evolution observed in our application, where the incumbent upper bound typically improves early in the algorithm, while later convergence is mainly driven by lower bound improvements.

Replacing the fixed parameter λ with an iteration-dependent coefficient λ^j, the level imposed in $(LS\text{-}M)$ is defined as:

$$\ell^j(\lambda^j) = \lambda^j \cdot LB^j + (1 - \lambda^j) \cdot (UB^{j-1})^* \tag{4}$$

Instead of using one progress ratio, as in [2], we update λ^j from two ratios. This separates the lower bound progress from the incumbent upper bound progress, which is useful because the two bounds may improve at different rates. The first ratio, r^j_L, compares the lower bound improvement with the gap between

[2] $j_{\max}$: iteration limit, ϵ_{tol}: gap tolerance, $\tau_{\max}$: elapsed time limit.

the previous level ℓ^{j-1} and the previous lower bound. The second ratio, r_U^j, compares the incumbent upper bound improvement with the gap between the previous upper bound and the previous level ℓ^{j-1}:

$$r_L^j = \min\left\{1,\ \frac{LB^j - LB^{j-1}}{\ell^{j-1} - LB^{j-1}}\right\}$$
$$r_U^j = \min\left\{1,\ \frac{(UB^{j-2})^* - (UB^{j-1})^*}{(UB^{j-2})^* - \ell^{j-1}}\right\}$$

Since the lower bound is nondecreasing, the incumbent upper bound is nonincreasing, and the previous level ℓ^{j-1} lies strictly between these bounds, the numerators and denominators in the raw progress ratios are nonnegative. The update routine then caps these ratios at one, so that r_L^j and r_U^j lie in $[0, 1]$.

The update of λ^j is based on the position of r_L^j and r_U^j relative to two thresholds, $r_{\min}$ and $r_{\max}$, with $0 < r_{\min} < r_{\max} < 1$, as illustrated in Figure 1. Values below $r_{\min}$ indicate insufficient progress, values between $r_{\min}$ and $r_{\max}$ indicate moderate progress, and values above $r_{\max}$ indicate satisfactory progress.

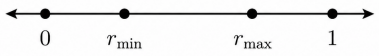


Figure 1: Interval partition induced by the thresholds $0 < r_{\min} < r_{\max} < 1$ in the dynamic level parameter update.

Compared with the dynamic parametrization routine in [2], the update used here includes not only a linear correction but also a more aggressive quadratic correction, which performed better in our application. When both ratios are below $r_{\min}$, Routine S2 applies the quadratic correction to change λ^j more strongly. When only one ratio is below $r_{\min}$ and the other remains between $r_{\min}$ and $r_{\max}$, it applies the milder linear correction. In the remaining cases, λ^j is left unchanged. The parameter $\gamma \in (0, 1)$ controls the size of the linear and quadratic corrections. After each update, λ^j is restricted to the interval $0 < \lambda_{\min} \le \lambda^j \le \lambda_{\max} < 1$. This keeps the level $\ell^j(\lambda^j)$ imposed in $(LS\text{-}M)$ away from both LB^j and $(UB^{j-1})^*$, and avoids numerical instabilities that may arise when the level constraint is placed too close to either bound.

Routine S2 Dynamic update of the level parameter λ^j

INPUT: λ^{j-1}, LB^j, LB^{j-1}, ℓ^{j-1}, $(UB^{j-1})^*$, $(UB^{j-2})^*$, γ, $r_{\min}$, $r_{\max}$, $\lambda_{\min}$, and $\lambda_{\max}$.
Compute r_L^j and r_U^j.
Set $\lambda^j \leftarrow \lambda^{j-1}$.
If $r_L^j < r_{\min}$ and $r_U^j < r_{\min}$:
  if $r_L^j < r_U^j$, set $\lambda^j \leftarrow 1 - (1 - \lambda^{j-1})\gamma^2$,
  otherwise, set $\lambda^j \leftarrow \lambda^{j-1}\gamma^2$.
If $r_L^j < r_{\min}$ and $r_{\min} \leq r_U^j \leq r_{\max}$, set $\lambda^j \leftarrow 1 - (1 - \lambda^{j-1})\gamma$.
If $r_U^j < r_{\min}$ and $r_{\min} \leq r_L^j \leq r_{\max}$, set $\lambda^j \leftarrow \lambda^{j-1}\gamma$.
Bound $\lambda^j \leftarrow \min\{\lambda_{\max}, \max\{\lambda^j, \lambda_{\min}\}\}$.

Algorithm S3 presents the full decomposition scheme that uses the dynamic parametrization introduced in Routine S2. The algorithm terminates when the iteration counter exceeds the limit $j_{\max}$, the relative optimality gap falls below the tolerance ϵ_{tol}, or the elapsed wall-clock time exceeds $\tau_{\max}$. The initial level parameter is denoted λ^1.

Algorithm S3 Benders Decomposition with Dynamic Level Set Parametrization

INPUT: λ^1, $j_{\max}$, ϵ_{tol}, $\tau_{\max}$, γ, $r_{\min}$, $r_{\max}$, $\lambda_{\min}$, and $\lambda_{\max}$.
INITIALIZE: Compute $\widehat{z}^0$ by solving the deterministic wait-and-see problems and averaging the resulting first-stage decisions. Set $\widehat{z} \leftarrow \widehat{z}^0$, initialize $(UB^0)^*$ with a sufficiently large value, set $j \leftarrow 1$ and $\epsilon^0 \leftarrow +\infty$.
while $j \leq j_{\max}$, $\epsilon^{j-1} > \epsilon_{\text{tol}}$, and elapsed time $< \tau_{\max}$ **do**
  Solve (M) and set LB^j equal to its objective value.
  If $j \geq 2$, update λ^j using Routine S2.
  Compute $\ell^j(\lambda^j)$ from (4).
  Solve $(LS\text{-}M)$ and obtain $\bar{z}^j = (\bar{x}^j, \bar{q}^j, \bar{v}^j)$.
  for $k \in \mathcal{K}$, $\omega \in \Omega$ **do**
    Solve $\mathcal{V}_{k,\omega}(\bar{z}^j)$ and generate $c_{k,\omega}^j$.
  end for
  Compute UB^j from (1) and set $(UB^j)^* \leftarrow \min\{(UB^{j-1})^*, UB^j\}$.
  If $(UB^j)^* < (UB^{j-1})^*$, set $\widehat{z} \leftarrow \bar{z}^j$.
  Set $\epsilon^j \leftarrow ((UB^j)^* - LB^j)/LB^j$.
  Add $c_{k,\omega}^j$, $k \in \mathcal{K}$, $\omega \in \Omega$, to (M) and $(LS\text{-}M)$.
  Set $j \leftarrow j + 1$.
end while

2.3 Managing the growth of the LP and QP master problems

This subsection complements Section III-E of the main paper [1] by discussing practical strategies to limit the growth of the master problems. In particular, it

describes the cut consolidation and cut sparsification procedures used to control the number and density of cuts added during the decomposition algorithm.

As the algorithm progresses, new cuts are added at each iteration to the LP and QP master problems. In the multi-cut setting adopted in this work, $|\mathcal{K}| \cdot |\Omega|$ cuts are appended per iteration. As a result, both master problems can grow rapidly and become increasingly expensive to solve. Several strategies have been proposed to keep the cut pool manageable.

In [3] the authors introduce an adaptive multicut aggregation scheme that partitions the scenario set into groups and dynamically reduces the number of groups during the algorithm, showing that intermediate aggregation levels can yield substantial computational savings.

In the asynchronous Benders decomposition framework of [4], the proposed cut management strategy does not add every cut generated by the workers to the master formulation. The master gives priority to cuts that are sufficiently violated at the current solution. When several candidate cuts correspond to the same recourse variable θ_s in the notation of [4], the most violated cut is retained. In our temporal multi-cut formulation, this corresponds to selecting among cuts associated with the same $\theta_{k,\omega}$ for a given segment k and scenario ω. The purpose is to keep the master problem smaller while preserving cuts that are most relevant for improving the current cutting-plane approximation.

In [5], the authors propose a cut consolidation technique for the nested L-shaped method. After the master problem is solved, the dual multiplier associated with each cut is inspected. Cuts with a zero dual multiplier are considered inactive, and the number of consecutive iterations for which each cut remains inactive is tracked. Cuts that remain inactive for sufficiently long can then be combined into an aggregated cut, reducing the size of the master problem while retaining part of the information contained in the removed cuts.

We adapt this idea to our temporal multi-cut setting. Each cut $c^{\nu}_{k,\omega}$ is associated with a temporal segment k, a scenario ω, and the iteration ν at which the cut was generated. For each fixed pair (k, ν), the cuts $\{c^{\nu}_{k,\omega}\}_{\omega \in \Omega}$ form one consolidation group.

The consolidation procedure is controlled by two parameters, N_c and R_c. Parameter N_c specifies the number of consecutive iterations for which a cut must remain inactive before it becomes eligible for consolidation. At iteration j, we denote the scenarios associated with such cuts by

$$\Omega^{j,\text{inact}}_{k,\nu} = \left\{\omega \in \Omega : c^{\nu}_{k,\omega} \text{ satisfies the } N_c \text{ inactivity criterion}\right\} \tag{5}$$

Parameter R_c specifies the minimum fraction of cuts in a consolidation group that must satisfy this inactivity criterion before consolidation is performed. Thus, consolidation is triggered when

$$\frac{\left|\Omega^{j,\text{inact}}_{k,\nu}\right|}{|\Omega|} \geq R_c \tag{6}$$

When this condition is satisfied, only the eligible inactive cuts, that is, those indexed by $\omega \in \Omega^{j,\text{inact}}_{k,\nu}$, are removed. They are replaced by the single aggregated

cut

$$\sum_{\omega \in \Omega_{k,\nu}^{j,\text{inact}}} \pi_\omega \theta_{k,\omega} \geq \sum_{\omega \in \Omega_{k,\nu}^{j,\text{inact}}} \pi_\omega c_{k,\omega}^{\nu} \left(x, q_{k,\omega}, v_{t_{k-1},\omega}, v_{t_k,\omega}\right) \tag{7}$$

The remaining cuts in the same consolidation group are kept individually in the master problems. In this way, the number of constraints is reduced without removing all the information carried by the inactive cuts.

Replacing several individual cuts by one aggregated cut weakens the cutting-plane approximation and may therefore cause the lower bound to decrease. To prevent this, we add the following lower bound preservation constraint to both master problems:

$$\sum_{n \in N,\, g \in G^{\text{INV}}(n)} IC_g x_g + \mathbb{E}_\omega \left[\sum_{k \in \mathcal{K}} \theta_{k,\omega} \right] \geq (LB^{j-1})^* \tag{8}$$

Here,

$$(LB^{j-1})^* = \max_{1 \leq n < j} LB^v \tag{9}$$

is the best lower bound obtained before iteration j. Therefore, constraint (8) prevents cut consolidation from reducing the lower bound below its best previously obtained value.

Moreover, we also apply a simple cut sparsification rule when forming the cuts. During cut construction, coefficients with absolute value at most 10^{-6} are set to zero. The purpose is to remove numerically negligible terms that contribute very little to the cut while making it denser. In practice, this retains only the meaningful coefficients in the cut description, resulting in sparser cut constraints and improved numerical conditioning in the LP and QP master problems.

2.4 Valid inequalities for the LP and QP master problems

This subsection complements Section III-E of the main manuscript [1], where valid inequalities are introduced as one of the enhancements of the proposed stabilized Benders decomposition algorithm. Here, we provide the explicit inequalities that are added to the LP and QP master problems. These inequalities are tailored to the renewable auction capacity expansion model and exploit structural properties of the second stage operating problem.

In the early iterations, before many Benders cuts have been generated, the LP and QP master problems provide only a coarse approximation of the second stage operating problem. To strengthen these master problems, we add problem-specific valid inequalities that are satisfied by every feasible solution of the original stochastic problem. These inequalities are constructed using only parameters and master problem variables, and therefore can be imposed directly in the master problems. We consider two types of inequalities. The first type is related to renewable production and capacity limits. The second type is related

to hydro storage dynamics, with separate bounds for pumped-storage and reservoir units. For completeness, we state the validity of each family of inequalities formally and provide the corresponding short proof.

• RES production and capacity bounds. The net renewable production over segment k is bounded above by the sum of existing and invested RES output, since production shedding $ps_{n,t,\omega} \geq 0$ can only reduce it. Introducing an auxiliary variable $\xi_{k,\omega} \geq 0$ and setting the penalty coefficient M_k equal to the slack penalty of the production target constraint, Eq. (10) bounds the excess of the target $q_{k,\omega}$ over the total RES capacity, and Eq. (11) converts this excess into a lower bound on $\theta_{k,\omega}$:

$$\xi_{k,\omega} \geq q_{k,\omega} - \sum_{\substack{n \in N \\ g \in G^{RE}(n)}} \sum_{t=t_{k-1}+1}^{t_k} P_g^{RE} \cdot TS_{g,t,\omega} - \sum_{\substack{n \in N \\ g \in G^{\mathrm{INV}}(n)}} \sum_{t=t_{k-1}+1}^{t_k} TS_{g,t,\omega} \cdot x_g \tag{10}$$

$$\theta_{k,\omega} \geq M_k \cdot \xi_{k,\omega} \qquad \forall k \in \mathcal{K},\, \omega \in \Omega \tag{11}$$

Proposition 1. *Constraints (10)–(11) are valid for the stochastic problem.*

Proof. Using the renewable generation constraints of the main paper [1]:

$$p_{g,t,\omega}^{RE} = P_g^{RE} \cdot TS_{g,t,\omega}, \quad g \in G^{RE}(n), \qquad p_{g,t,\omega}^{xRE} = x_g \cdot TS_{g,t,\omega}, \quad g \in G^{\mathrm{INV}}(n).$$

Since production shedding satisfies $ps_{n,t,\omega} \geq 0$, the net renewable output over segment k is bounded by:

$$\sum_{t=t_{k-1}+1}^{t_k} \left(\sum_{\substack{n \in N \\ g \in G^{RE}(n)}} p_{g,t,\omega}^{RE} + \sum_{\substack{n \in N \\ g \in G^{\mathrm{INV}}(n)}} p_{g,t,\omega}^{xRE} - \sum_{n \in N} ps_{n,t,\omega} \right)$$

$$\leq \sum_{\substack{n \in N \\ g \in G^{RE}(n)}} \sum_{t=t_{k-1}+1}^{t_k} P_g^{RE} \cdot TS_{g,t,\omega} + \sum_{\substack{n \in N \\ g \in G^{\mathrm{INV}}(n)}} \sum_{t=t_{k-1}+1}^{t_k} TS_{g,t,\omega} \cdot x_g.$$

The budget constraint of the subproblem requires that the left hand side is at least $q_{k,\omega}$. Defining:

$$\xi_{k,\omega} := \max\left\{0,\, q_{k,\omega} - \sum_{\substack{n \in N \\ g \in G^{RE}(n)}} \sum_{t=t_{k-1}+1}^{t_k} P_g^{RE} \cdot TS_{g,t,\omega} - \sum_{\substack{n \in N \\ g \in G^{\mathrm{INV}}(n)}} \sum_{t=t_{k-1}+1}^{t_k} TS_{g,t,\omega} \cdot x_g \right\}$$

satisfies (10). When $\xi_{k,\omega} > 0$, the net renewable production budget exceeds the available RES capacity and the subproblem incurs a penalty of at least $M_k \cdot \xi_{k,\omega}$, so $\theta_{k,\omega} \geq M_k \cdot \xi_{k,\omega}$, which is (11). □

• Pumped storage intertemporal bounds. Let $T_k = |t_k - t_{k-1}|$ denote the duration of segment k. For pumped-storage units $g \in H^{PS}(n)$, the storage level can increase by pumping and decrease by generating:

$$v^{PS}_{g,t_k,\omega} \geq v^{PS}_{g,t_{k-1},\omega} - \frac{T_k \cdot Q^{PS}_g}{\eta_g} \tag{12}$$

$$v^{PS}_{g,t_k,\omega} \leq v^{PS}_{g,t_{k-1},\omega} + T_k \cdot D_g \cdot \eta_p \tag{13}$$

$$\forall g \in H^{PS}(n),\, n \in N,\, k \in \mathcal{K},\, \omega \in \Omega$$

Proposition 2. *Constraints (12)–(13) are valid for the stochastic problem.*

Proof. From the storage balance presented in the main paper [1], the pumped storage level satisfies:

$$v^{PS}_{g,t,\omega} = v^{PS}_{g,t-1,\omega} + \eta_p \cdot d_{g,t,\omega} - \frac{q^{PS}_{g,t,\omega}}{\eta_g}$$

with $0 \leq d_{g,t,\omega} \leq D_g$ and $0 \leq q^{PS}_{g,t,\omega} \leq Q^{PS}_g$ Summing from $t_{k-1}+1$ to t_k:

$$v^{PS}_{g,t_k,\omega} = v^{PS}_{g,t_{k-1},\omega} + \sum_{t=t_{k-1}+1}^{t_k} \left(\eta_p \, d_{g,t,\omega} - \frac{q^{PS}_{g,t,\omega}}{\eta_g} \right)$$

The upper bound follows from $d_{g,t,\omega} \leq D_g$ and $q^{PS}_{g,t,\omega} \geq 0$:

$$v^{PS}_{g,t_k,\omega} \leq v^{PS}_{g,t_{k-1},\omega} + T_k \cdot D_g \cdot \eta_p$$

The lower bound follows from $d_{g,t,\omega} \geq 0$ and $q^{PS}_{g,t,\omega} \leq Q^{PS}_g$:

$$v^{PS}_{g,t_k,\omega} \geq v^{PS}_{g,t_{k-1},\omega} - \frac{T_k \cdot Q^{PS}_g}{\eta_g}$$

□

• Reservoir inflow bounds. For reservoir units $g \in H^R(n)$, spillage $s_{g,t,\omega} \geq 0$ can only decrease the storage level, so only an upper bound is valid:

$$v^{R}_{g,t_k,\omega} \leq v^{R}_{g,t_{k-1},\omega} + \sum_{t=t_{k-1}+1}^{t_k} A^{R}_{g,t,\omega} \tag{14}$$

$$\forall g \in H^{R}(n),\, n \in N,\, k \in \mathcal{K},\, \omega \in \Omega$$

Proposition 3. *Constraint (14) is valid for the stochastic problem.*

Proof. From the reservoir balance presented in the main paper [1]:

$$v^R_{g,t,\omega} = v^R_{g,t-1,\omega} + A^R_{g,t,\omega} - \frac{q^R_{g,t,\omega}}{\eta_g} - s_{g,t,\omega}$$

with $q^R_{g,t,\omega} \geq 0$ and $s_{g,t,\omega} \geq 0$. Summing from $t_{k-1}+1$ to t_k:

$$v^R_{g,t_k,\omega} = v^R_{g,t_{k-1},\omega} + \sum_{t=t_{k-1}+1}^{t_k} \Big(A^R_{g,t,\omega} - \frac{q^R_{g,t,\omega}}{\eta_g} - s_{g,t,\omega} \Big)$$

Since $q^R_{g,t,\omega} \geq 0$ and $s_{g,t,\omega} \geq 0$:

$$v^R_{g,t_k,\omega} \leq v^R_{g,t_{k-1},\omega} + \sum_{t=t_{k-1}+1}^{t_k} A^R_{g,t,\omega}$$

giving (14). □

Together, Eqs. (10)–(14) incorporate subproblem structure into the LP and QP master problems, tighten the relaxation, and reduce the likelihood of selecting trial points that would require costly penalized slack variables in the subproblems. After adding the valid inequalities (10)–(14) and the lower bound preservation constraint (8), the LP and QP master problems can be written as follows:

$$\begin{aligned}
\text{(VC-M)} \quad \min_{x,q,v,\theta,\xi} \quad & \sum_{n\in N,\, g\in G^{\mathrm{INV}}(n)} IC_g \cdot x_g + \mathbb{E}_\omega\Big[\sum_{k\in\mathcal{K}} \theta_{k,\omega}\Big] \\
\text{s.t.} \quad & x_g \leq X_g \quad \forall g \in G^{\mathrm{INV}}(n),\, n \in N \\
& \mathbb{E}_\omega\Big[\textstyle\sum_{k\in\mathcal{K}} q_{k,\omega}\Big] = TV \\
& (8)\text{–}(14) \\
& v^R_{g,t_k,\omega} \leq V^R_g, \quad v^{PS}_{g,t_k,\omega} \leq V^{PS}_g \quad \forall k \in \mathcal{K},\, \omega \in \Omega \\
& \theta_{k,\omega} \geq c^\nu_{k,\omega}(x, q_{k,\omega}, v_{t_{k-1},\omega}, v_{t_k,\omega}) \\
& \quad \forall \nu \in \{1,\ldots,j-1\},\, k \in \mathcal{K},\, \omega \in \Omega \\
& \xi_{k,\omega} \geq 0 \quad \forall k \in \mathcal{K},\, \omega \in \Omega
\end{aligned}$$

$$
\begin{aligned}
\text{(VC-LS-M)} \quad \min_{x,q,v,\theta,\xi} \ & \|z - \widehat{z}\|_2^2 \\
\text{s.t. } & x_g \le X_g \quad \forall g \in G^{\mathrm{INV}}(n),\, n \in N \\
& \mathbb{E}_\omega\Big[\textstyle\sum_{k\in\mathcal{K}} q_{k,\omega}\Big] = TV \\
& \sum_{n\in N,\, g\in G^{\mathrm{INV}}(n)} IC_g \cdot x_g + \mathbb{E}_\omega\Big[\textstyle\sum_{k\in\mathcal{K}} \theta_{k,\omega}\Big] \le \ell^j(\lambda^j) \\
& (8)\text{–}(14) \\
& v^R_{g,t_k,\omega} \le V^R_g, \quad v^{PS}_{g,t_k,\omega} \le V^{PS}_g \quad \forall k \in \mathcal{K},\, \omega \in \Omega \\
& \theta_{k,\omega} \ge c^\nu_{k,\omega}(x, q_{k,\omega}, v_{t_{k-1},\omega}, v_{t_k,\omega}) \\
& \quad \forall \nu \in \{1,\dots,j-1\},\, k \in \mathcal{K},\, \omega \in \Omega \\
& \xi_{k,\omega} \ge 0 \quad \forall k \in \mathcal{K},\, \omega \in \Omega
\end{aligned}
$$

With these strengthened master problems in place, the proposed decomposition algorithm is obtained by combining dynamic level parametrization, cut sparsification, cut consolidation, and the valid inequalities above. Algorithm S4 collects the steps of the proposed decomposition algorithm presented as Algorithm 1 in the main paper. The valid inequalities and the lower bound preservation constraint are embedded in (*VC-M*) and (*VC-LS-M*), while Routine S2 updates the level parameter.

Algorithm S4 Proposed Decomposition Algorithm

INPUT: λ^1, $j_{\max}$, ϵ_{tol}, $\tau_{\max}$, γ, $r_{\min}$, $r_{\max}$, $\lambda_{\min}$, $\lambda_{\max}$, N_c, and R_c.
INITIALIZE: Compute $\widehat{z}^0$ by solving the deterministic wait-and-see problems and averaging the resulting first stage decisions. Set $\widehat{z} \leftarrow \widehat{z}^0$, initialize $(UB^0)^*$ with a sufficiently large value, set $j \leftarrow 1$ and $\epsilon^0 \leftarrow +\infty$.
while $j \le j_{\max}$, $\epsilon^{j-1} > \epsilon_{\mathrm{tol}}$, and elapsed time $< \tau_{\max}$ **do**
  Solve (*VC-M*) and set LB^j equal to its objective value.
  If $j \ge 2$, update λ^j with Routine S2.
  Compute $\ell^j(\lambda^j)$ from (4).
  Solve (*VC-LS-M*) and obtain $\bar{z}^j = (\bar{x}^j, \bar{q}^j, \bar{v}^j)$.
  for $k \in \mathcal{K}$, $\omega \in \Omega$ **do**
    Solve $\mathcal{V}_{k,\omega}(\bar{z}^j)$ and generate $c^j_{k,\omega}$.
  end for
  Compute UB^j from (1) and set $(UB^j)^* \leftarrow \min\{(UB^{j-1})^*, UB^j\}$.
  If $(UB^j)^* < (UB^{j-1})^*$, set $\widehat{z} \leftarrow \bar{z}^j$.
  Set $\epsilon^j \leftarrow ((UB^j)^* - LB^j)/LB^j$.
  Add $c^j_{k,\omega}$, $k \in \mathcal{K}$, $\omega \in \Omega$, to (*VC-M*) and (*VC-LS-M*).
  Consolidate inactive cuts using N_c and R_c.
  Set $j \leftarrow j + 1$.
end while

3 CWE case study

This section presents additional computational results for the Central Western European (CWE) subsystem that complement Section V-E of the main paper [1]. We first present the CWE generation fleet and dataset, then investigate the sensitivity of the proposed decomposition algorithm (Algorithm S4, which corresponds to Algorithm 1 of the main paper, as discussed in Section 2) to the number of temporal segments, compare the fixed parameter level set algorithm (Algorithm S1), the dynamic level set algorithm (Algorithm S3), the proximal bundle method, and the proposed decomposition algorithm, and finally perform a sensitivity analysis with respect to the number of scenarios and the target RES volume, including the corresponding expected value of perfect information.

The CWE subsystem comprises 7 countries: Austria, Belgium, France, Germany, Luxembourg, the Netherlands, and Switzerland, representing approximately 40% of the full-Eu dataset described in Section V of the main paper [1]. Table 1 provides an overview of the CWE dataset.

Table 1: Overview of the CWE dataset

	Characteristics	Data source
Geography	- CWE subsystem 7 countries	
Transmission network	- Nodal model - 1549 buses, - 2222 AC lines, 15 DC lines	PyPSA [6]
Thermal generation	- Unit-based model - 918 generation units	PyPSA
Hydrological generation	- Unit-based model - 3 technologies: run-of-river, reservoir, pumped storage - 121 reservoirs, 66 pumped storage units, 367 run-of-river units	PyPSA
Solar & wind generation	- 1546 solar units, 1318 onshore wind, 99 offshore wind units	PyPSA
Candidate RES investments	- 261 solar, 254 solar hsat, 203 onshore wind, 16 offshore wind projects	PyPSA
Uncertainty	- 15 scenarios - Each scenario consists of a yearly demand, inflow, solar and wind profiles	PyPSA ERAA [7]
Time horizon	- One operational year with 6 time periods per day	

Figure 2 provides a geographic view of the existing generation fleet before investment. Each bus is represented by a pie chart, the size of which is proportional to the total installed capacity at that location, with slices colored by technology. Transmission lines are drawn between buses with line thickness proportional to the thermal capacity.

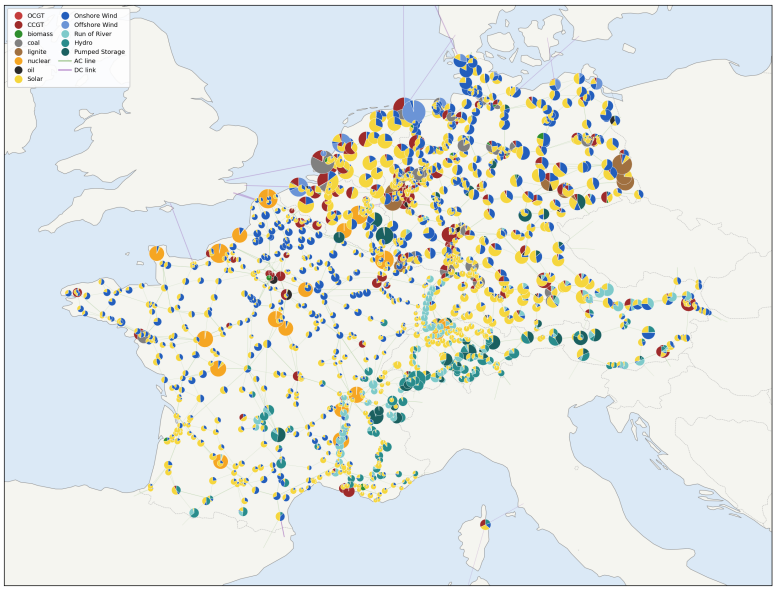


Figure 2: Existing generation fleet of the CWE subsystem before investment.

3.1 Sensitivity to the number of temporal segments

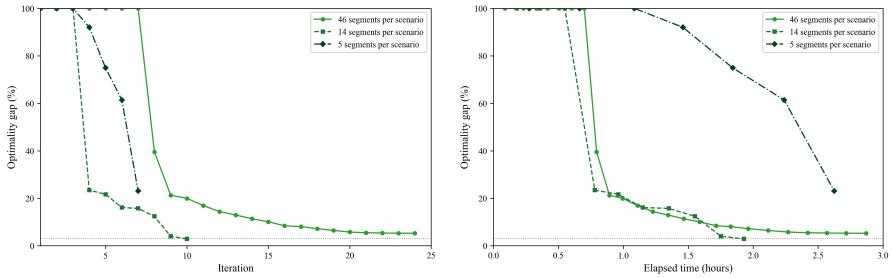
The temporal decomposition partitions each scenario into a set of temporal segments, and each temporal segment defines one subproblem. The number of temporal segments governs the granularity of the decomposition. Increasing the number of temporal segments produces smaller and cheaper subproblems that can be distributed across more cores, thereby reducing the subproblem solve time per iteration. However, this also increases the number of variables and constraints in the LP and QP master problems and adds many more cuts per iteration, which accelerates the growth of the master problems and can eventually render them intractable. Conversely, fewer temporal segments yield larger subproblems that are more expensive to solve individually, but the LP and QP master problems remain smaller and grow less rapidly. A tradeoff therefore exists between subproblem parallelism and the tractability of the LP and QP master problems.

To investigate this tradeoff, Figure 3 compares the proposed decomposition algorithm (Algorithm S4) for three different configurations on the CWE instance with 15 scenarios, namely 5, 14, and 46 temporal segments per scenario, corresponding to 75, 210, and 690 segment-scenario subproblems respectively. Panels 3a and 3b report the optimality gap over iterations and elapsed time. For the configuration with 75 subproblems, the runs are performed using 75 cores distributed across 3 computational nodes (25 cores and 400 GB of RAM per node). For the configurations with 210 and 690 subproblems, the runs use 108 cores distributed across 2 computational nodes (54 cores and 650 GB of

RAM per node).

With 14 temporal segments per scenario, the algorithm reaches a 3% optimality gap in less than 2 hours and is the only configuration to do so within the 3-hour time limit. The coarser partition with 5 temporal segments per scenario produces fewer but substantially larger subproblems, so that each iteration takes considerably longer and the algorithm does not reach the 3% threshold within the available time. The finer partition with 46 temporal segments per scenario converges more slowly as well, ending the 3-hour window with an optimality gap of approximately 5%, as the large number of cuts added per iteration accelerates the growth of the LP and QP master problems without a proportional gain in subproblem information.

These results support the choice of 14 temporal segments adopted in Section V-A of the main paper [1]. At this granularity, the LP and QP master problems remain manageable while the subproblem parallelism is still sufficient to keep the iterations fast. This same balance motivated the choice of 108 cores for the full-EU instance.



(a) Optimality gap evolution over iterations. (b) Optimality gap evolution over elapsed time.

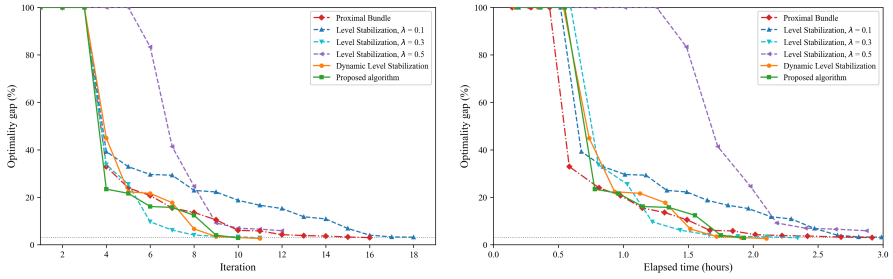
Figure 3: Convergence of the proposed decomposition algorithm (Algorithm S4) for 5, 14, and 46 temporal segments per scenario on the CWE instance with 15 scenarios. The gray dotted line marks the 3% optimality gap.

3.2 Algorithm comparison

We compare the proposed decomposition algorithm (Algorithm S4) against the dynamic level set algorithm (Algorithm S3), the fixed parameter level set algorithm (Algorithm S1) with three values of the fixed level parameter tested ($\lambda = 0.1$, $\lambda = 0.3$, and $\lambda = 0.5$), and a proximal bundle method. All methods share the same initialization strategy and are run within a 3-hour time window on 108 cores distributed across two computational nodes, each with 650 GB of RAM.

Figure 4 reports the optimality gap evolution over iterations and elapsed time in Panels 4a and 4b, respectively. The results in Figure 4 show that the fixed parameter level set scheme is sensitive to the choice of λ. Among the fixed

λ variants, $\lambda = 0.3$ performs best, while $\lambda = 0.1$ converges more slowly and $\lambda = 0.5$ performs worst, failing to reach the 3% gap threshold within the 3-hour time limit. Algorithms S3 and S4 avoid this manual choice by updating the level parameter dynamically. In both cases, we use $\gamma = 0.7$, $\lambda^1 = 0.1$, $r_{\min} = 0.1$, $r_{\max} = 0.9$, $\lambda_{\min} = 0.01$, and $\lambda_{\max} = 0.99$. Algorithm S4 further augments the dynamic level set scheme with the valid inequalities of Section 2.4.[3] As is evident from the figures, the proposed decomposition algorithm outperforms the alternatives and was therefore chosen for the full-EU experiments presented in the main paper.



(a) Optimality gap evolution over iterations. (b) Optimality gap evolution over elapsed time.

Figure 4: Convergence comparison of the proposed decomposition algorithm (Algorithm S4), the dynamic level set algorithm (Algorithm S3), the fixed parameter level set algorithm (Algorithm S1) with three fixed values of λ, and the proximal bundle method on the CWE instance. The gray dotted line marks the 3% optimality gap.

3.3 Sensitivity to the number of scenarios and the target RES volume

This subsection examines how the expected value of perfect information (EVPI) changes with the number of scenarios and with the target RES volume gap $\Delta TV = TV - TV_0$, where TV is the RES target and $TV_0 \approx 483$ TWh is the baseline target volume. For the experiments shown in this subsection, $\Delta TV \approx 17$ TWh.

We compute the EVPI as EVPI $= (C_{\text{SP}} - C_{\text{WS}})/C_{\text{WS}}$, where C_{SP}[4] is the total cost of the stochastic problem and C_{WS} is the average wait-and-see cost over the corresponding scenario set.

Figure 5 summarizes the EVPI sensitivity analysis. Panel 5a shows the EVPI as the number of scenarios increases from 5 to 15, with the target volume gap

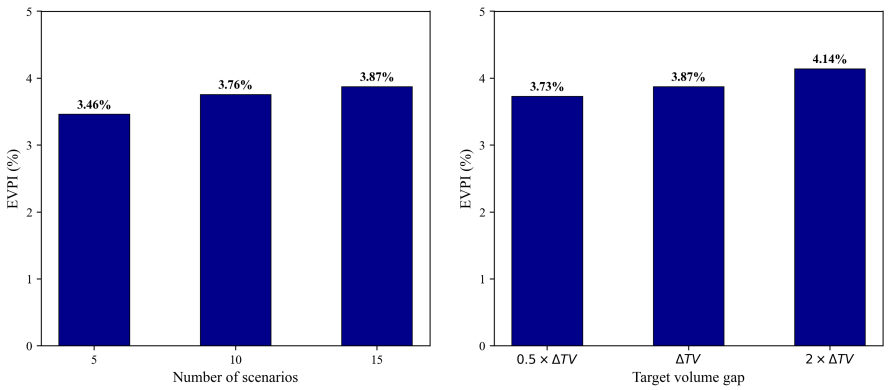
[3]The cut consolidation parameters were set to $N_c = 5$ and $R_c = 95\%$. However, cut consolidation was not triggered in this test, because convergence was reached after only 10 iterations and the inactivity conditions required for consolidation were not met.

[4]For the stochastic problem, the reported cost is equal to the best upper bound.

fixed at ΔTV. EVPI rises from 3.46% to 3.87%.

Panel 5b shows the EVPI as the target volume gap varies from $0.5 \times \Delta TV$ to $2 \times \Delta TV$, with the number of scenarios fixed at 15. The EVPI increases from 3.73% to 4.14%.

Taken together, these two sensitivities show that the cost of uncertainty becomes more pronounced when the problem becomes harder, either because the scenario set is larger or because the renewable target is more demanding.



(a) EVPI for an increasing number of scenarios.

(b) EVPI for an increasing target volume gap.

Figure 5: Sensitivity of the expected value of perfect information (EVPI) to the number of scenarios and the target RES volume gap on the CWE instance.

Tables 2 and 3 report the corresponding cost, invested capacity, and RES production for the stochastic problem (SP) and for the scenario-averaged wait-and-see benchmark (WS). In the WS columns, the costs are averages of the deterministic costs across scenarios, while the capacity and production values are averages of the corresponding deterministic decisions. In both tables, the EVPI is computed from the total costs defined above.

Table 2: Cost, invested capacity, and production breakdowns for different numbers of scenarios on the CWE instance with $\Delta TV \approx 17$ TWh.

	5 scenarios		10 scenarios		15 scenarios	
	SP	WS	SP	WS	SP	WS
Cost (billion €)						
Economic dispatch	27.24	28.06	27.09	27.90	27.18	27.49
Annualized investment	7.22	5.24	7.20	5.15	6.44	4.87
Total	34.45	33.30	34.30	33.05	33.62	32.36
Invested capacity (GW)						
Solar	42.6	29.7	41.6	29.1	38.3	25.9
Wind	41.9	35.3	40.1	34.7	38.2	33.5
Total	84.6	65.0	81.7	63.8	76.5	59.3
RES production (TWh)						
Total solar	241.3	222.6	241.9	223.1	236.9	219.0
Total wind	396.0	387.7	392.3	385.3	391.6	385.1
Production shedding	141.3	114.3	136.8	111.0	128.3	103.9
Net RES	496.0	496.0	497.4	497.4	500.2	500.2
EVPI (%)	3.46		3.76		3.87	

Table 3: Cost, invested capacity, and production breakdowns for different target volume gaps on the CWE instance with 15 scenarios.

	$0.5 \times \Delta TV$		$1 \times \Delta TV$		$2 \times \Delta TV$	
	SP	WS	SP	WS	SP	WS
Cost (billion €)						
Economic dispatch	27.60	27.61	27.18	27.49	27.03	27.25
Annualized investment	6.10	4.87	6.44	4.87	6.42	4.88
Total	33.70	32.48	33.62	32.36	33.45	32.12
Invested capacity (GW)						
Solar	35.3	25.9	38.3	25.9	49.7	25.9
Wind	39.5	33.4	38.2	33.5	39.2	33.5
Total	74.8	59.3	76.5	59.3	88.9	59.4
RES production (TWh)						
Total solar	232.4	219.0	236.9	219.0	252.5	219.1
Total wind	393.8	385.1	391.6	385.1	392.7	385.1
Production shedding	134.5	112.4	128.3	103.9	128.1	86.9
Net RES	491.7	491.7	500.2	500.2	517.1	517.3
EVPI (%)	3.73		3.87		4.14	

In the scenario sensitivity shown in Table 2, both the SP cost and the average WS cost decrease as more scenarios are included. The stochastic problem still invests more capacity than the average WS benchmark because it must remain feasible across all scenarios. As the number of scenarios increases, this hedging becomes less conservative in aggregate, with SP capacity falling from 84.6 GW to 76.5 GW and production shedding falling from 141.3 TWh to 128.3 TWh.

Over the same range, the EVPI rises from 3.46% to 3.87%, which indicates that the cost of making decisions before uncertainty is revealed becomes slightly larger when uncertainty is represented more fully.

In the target volume sensitivity analysis which is presented in Table 3, a higher RES target leads to more investment, especially in solar, and total SP capacity increases from 74.8 GW to 88.9 GW. The EVPI rises from 3.73% to 4.14% as ΔTV grows, which indicates that uncertainty matters more when the target becomes more demanding. The two sensitivities therefore point towards the same direction. A richer scenario representation and a larger RES requirement both increase the value of perfect information.

3.4 Renewable investments

Figures 6 and 7 present the renewable investments in the stochastic problem and in the average wait-and-see solution. Existing capacity is omitted from these figures to make the investment pattern easier to read. The existing generation fleet is shown separately in Figure 2.

Figure 6 presents the renewable investments in the stochastic problem solution. Compared with the average wait-and-see benchmark, these investments are more spatially concentrated.

Figure 7 presents the corresponding renewable investments for the average wait-and-see benchmark. They are more dispersed across the network, although the total invested capacity is smaller.

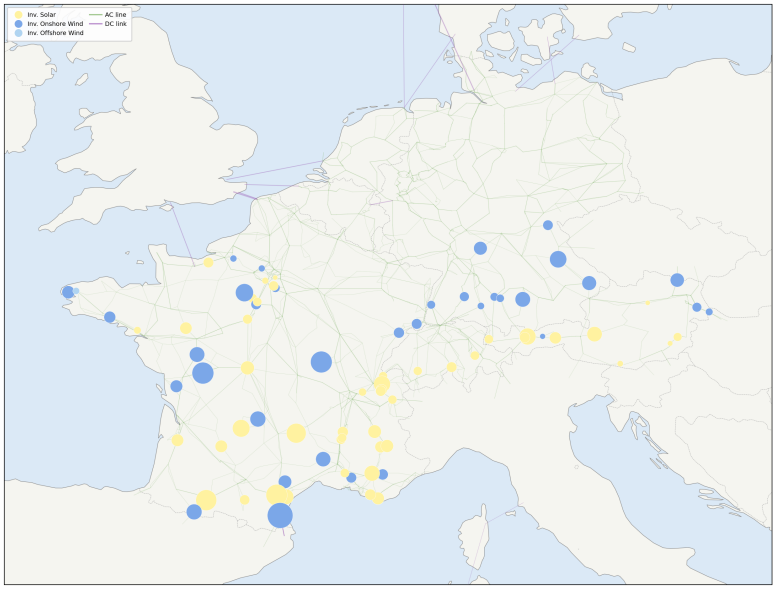


Figure 6: Renewable investments in the CWE subsystem for the stochastic problem solution.

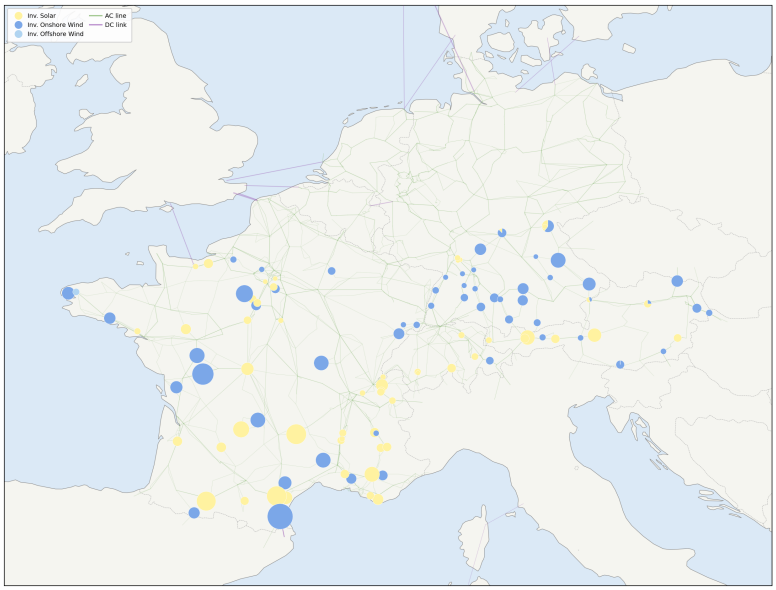


Figure 7: Renewable investments in the CWE subsystem for the average wait-and-see solution.

4 Full EU instance

This section reports additional results for the full EU instance discussed in Section V of the main paper. We first summarize the stochastic solution, then examine the synchronization overhead, and finally present the renewable investments obtained in the full-EU stochastic problem solution.

Figure 8 presents the existing generation fleet across the European network before investment.

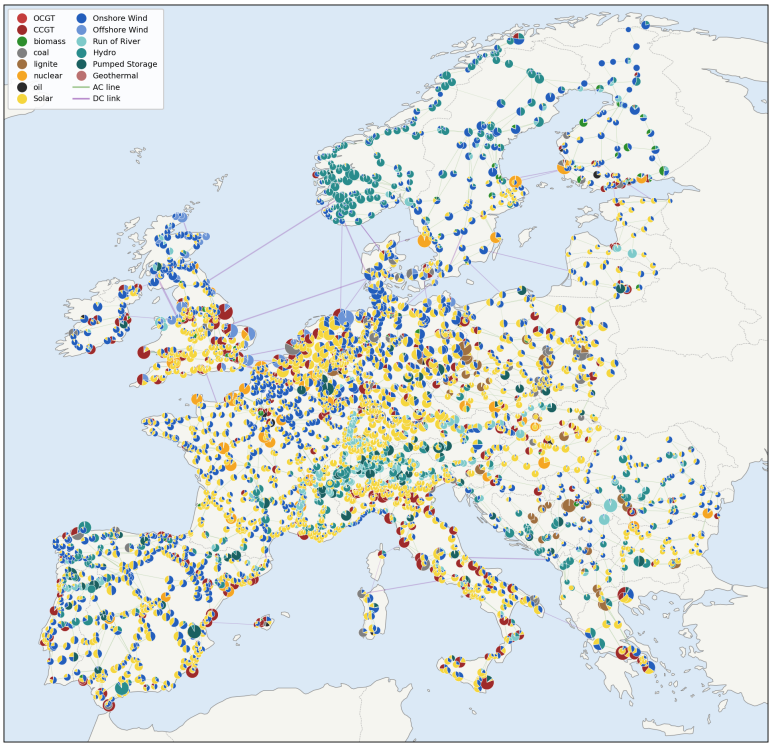


Figure 8: Existing generation fleet of the full EU instance before investment.

4.1 Stochastic problem results

Table 4 reports the cost, invested capacity, and production breakdowns for the stochastic problem. The algorithm stops after 49 iterations with an optimality gap of about 3%. The total annualized cost is 70.80 billion €. Of this amount, 5.10 billion € corresponds to annualized investment and the remaining 65.70 billion € to economic dispatch. The model installs 109.4 GW of new renewable capacity. Most of this capacity is solar, while wind accounts for the rest.

Table 4: Cost, invested capacity, and production breakdowns for the stochastic problem on the full EU instance with 15 scenarios.

	SP
Cost (billion €)	
Economic dispatch	65.70
Annualized investment	5.10
Total	70.80
Invested capacity (GW)	
Solar	74.7
Wind	34.6
Total	109.4
RES production (TWh)	
Total solar	473.0
Total wind	775.9
Production shedding	157.0
Net RES	1091.9
Optimality gap (%)	3.0

4.2 Synchronization overhead in temporal decomposition

Temporal decomposition introduces a synchronization barrier at every iteration because the next master problem solve can start only after all workers have completed their assigned subproblems. For iteration j, we measure the relative synchronization overhead as

$$\eta^j = \frac{T^j_{\max} - T^j_{\min}}{T^j_{\max}} \cdot 100\%,$$

where $T^j_{\min}$ and $T^j_{\max}$ denote the earliest and latest worker completion times during the subproblem solving phase. The denominator is therefore the completion time of the slowest worker in that iteration. Figure 9 shows the distribution of η^j across all iterations. The median overhead is about 79%, with most values between 75% and 83%, indicating substantial heterogeneity in worker completion times. This motivates asynchronous parallelization and improved load-balancing strategies for allocating subproblems to workers.

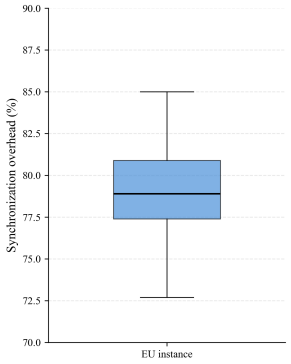


Figure 9: Distribution of relative synchronization overhead per iteration for the full EU instance.

4.3 Renewable investments

Figure 10 presents the renewable investments in the full-EU stochastic problem solution. Existing capacity is omitted from this figure to make the investment pattern easier to read, while the existing generation fleet is shown separately in Figure 8. The largest investments are visible in France, Scandinavia, and parts of southern Europe, mostly in locations where storage facilities already exist.

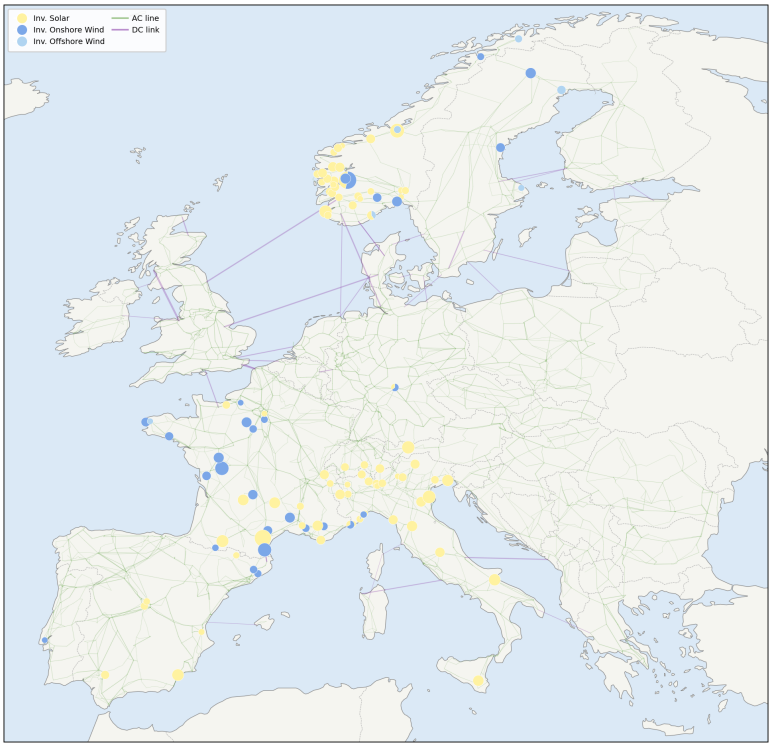


Figure 10: Renewable investments in the full EU instance for the stochastic problem solution.

References

[1] A. Visas, D. Ávila, A. Papavasiliou, G. Thomassen, and D. Pozo, "Decomposition techniques for model-based bid selection in renewable auctions," 2026.

[2] H. Zhang, I. E. Grossmann, K. McKinnon, B. R. Knudsen, R. G. Nava, and A. Tomasgard, "Integrated investment, retrofit and abandonment energy system planning with multi-timescale uncertainty using stabilised adaptive Benders decomposition," *European Journal of Operational Research*, vol. 325, no. 2, pp. 261–280, 2025.

[3] S. Trukhanov, L. Ntaimo, and A. Schaefer, "Adaptive multicut aggregation for two-stage stochastic linear programs with recourse," *European Journal of Operational Research*, vol. 206, no. 2, pp. 395–406, 2010.

[4] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei, "An asynchronous parallel benders decomposition method for stochastic network design problems," *Computers & Operations Research*, vol. 162, p. 106459, 2024.

[5] C. Wolf and A. Koberstein, "Dynamic sequencing and cut consolidation for the parallel hybrid-cut nested l-shaped method," *European Journal of Operational Research*, vol. 230, no. 1, pp. 143–156, 2013.

[6] T. Brown, J. Hörsch, and D. Schlachtberger, "PyPSA: Python for Power System Analysis," *Journal of Open Research Software*, vol. 6, no. 1, p. 4, Jan. 2018.

[7] ENTSO-e, "ERAA 2021 Modelling data." [Online]. Available: https://www.entsoe.eu/eraa/2021/modelling-data/