An EU-wide electricity generation capacity expansion problem subject to network constraints

Anthony Papavasiliou, National Technical University of Athens (NTUA), Greece

November 28, 2025

Energy System Modeling Workshop

TU Graz, Austria

Outline

- Context
  - Renewable energy integration
  - Capacity markets
  - Capacity expansion planning in the EU
- Power flow
- Capacity expansion model
- Algorithmics
  - Decoupling the target volume constraint
  - Scenario decomposition
  - Regularization
  - Temporal decomposition
  - Parallel computing implementation
- Results
- Ongoing Extensions
- Conclusions

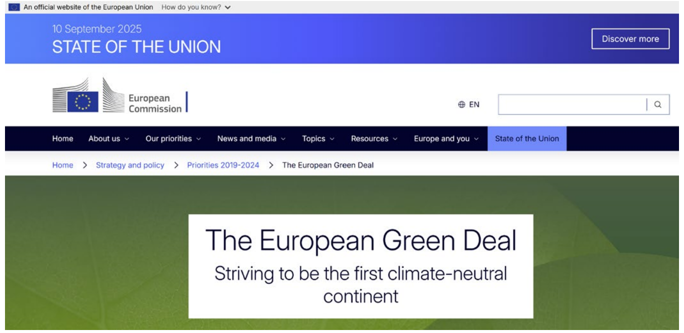
Context

Renewable energy integration

Capacity markets

Capacity expansion planning in the EU

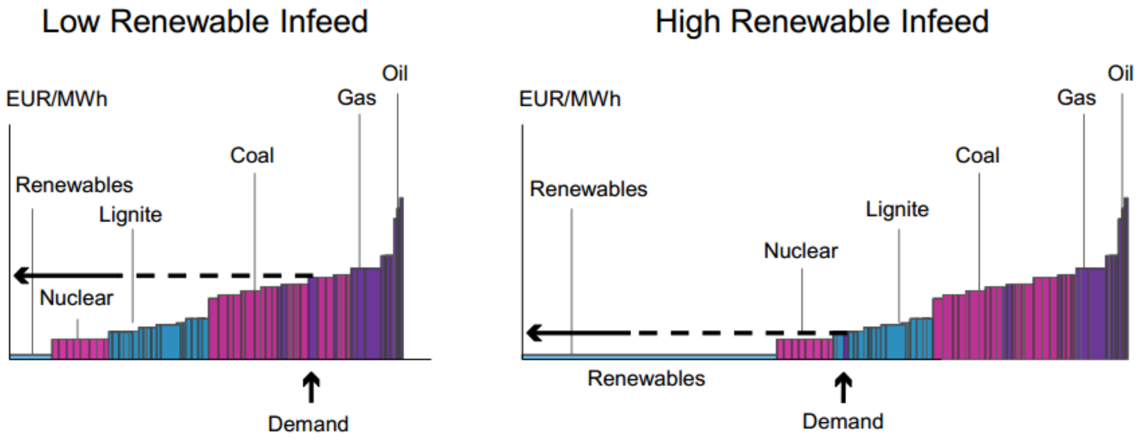
European Green Deal

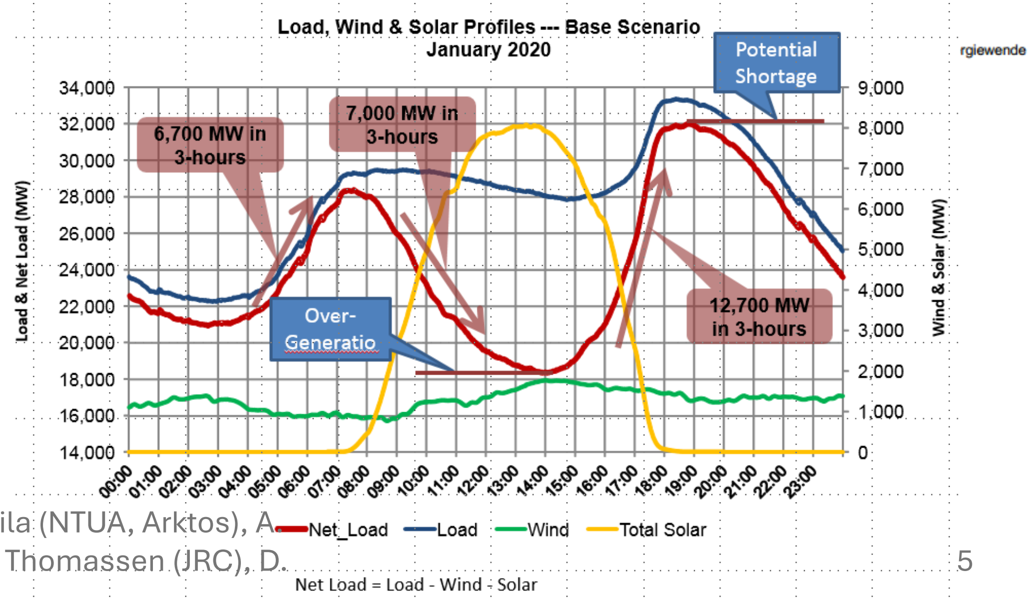


- New terms for the European green deal in 2019 [1]
  - Goal: carbon neutrality by 2050
  - 2030 goal: 55% greenhouse gas emissions reductions from 1990 levels
  - Original target for was 40%, set in 2014
- EU goal: share of 38-40% from renewable resources in the European electricity mix by 2030
  - In 2018, the share of renewable resources represented 18% of annual energy consumption [2], still long way to go
- Between 2010 and 2018, solar and wind capacity grew from 110 GW to 261 GW
  - Accelerated by European policy incentives
  - Cost decline of wind and solar electricity by 75% [2]

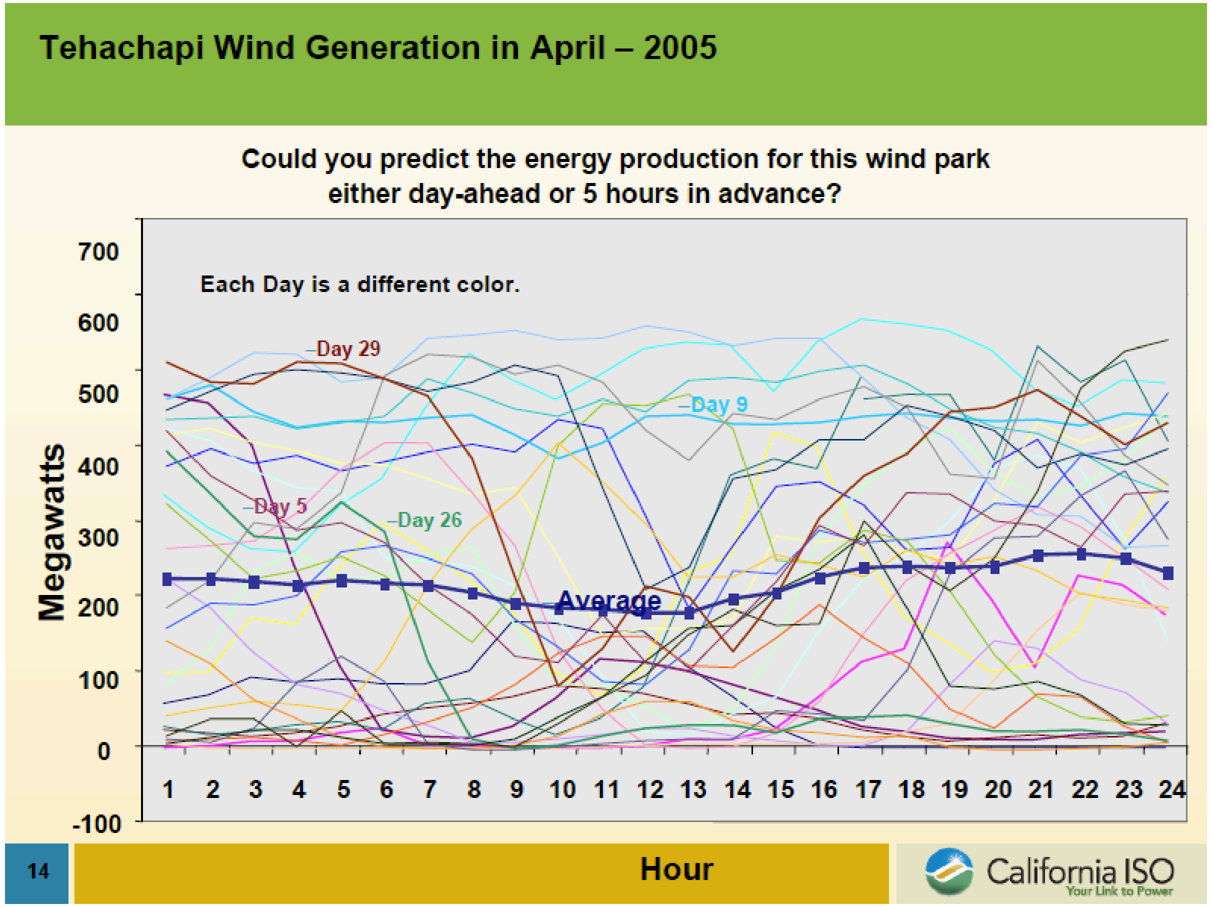
Challenges of renewable energy integration

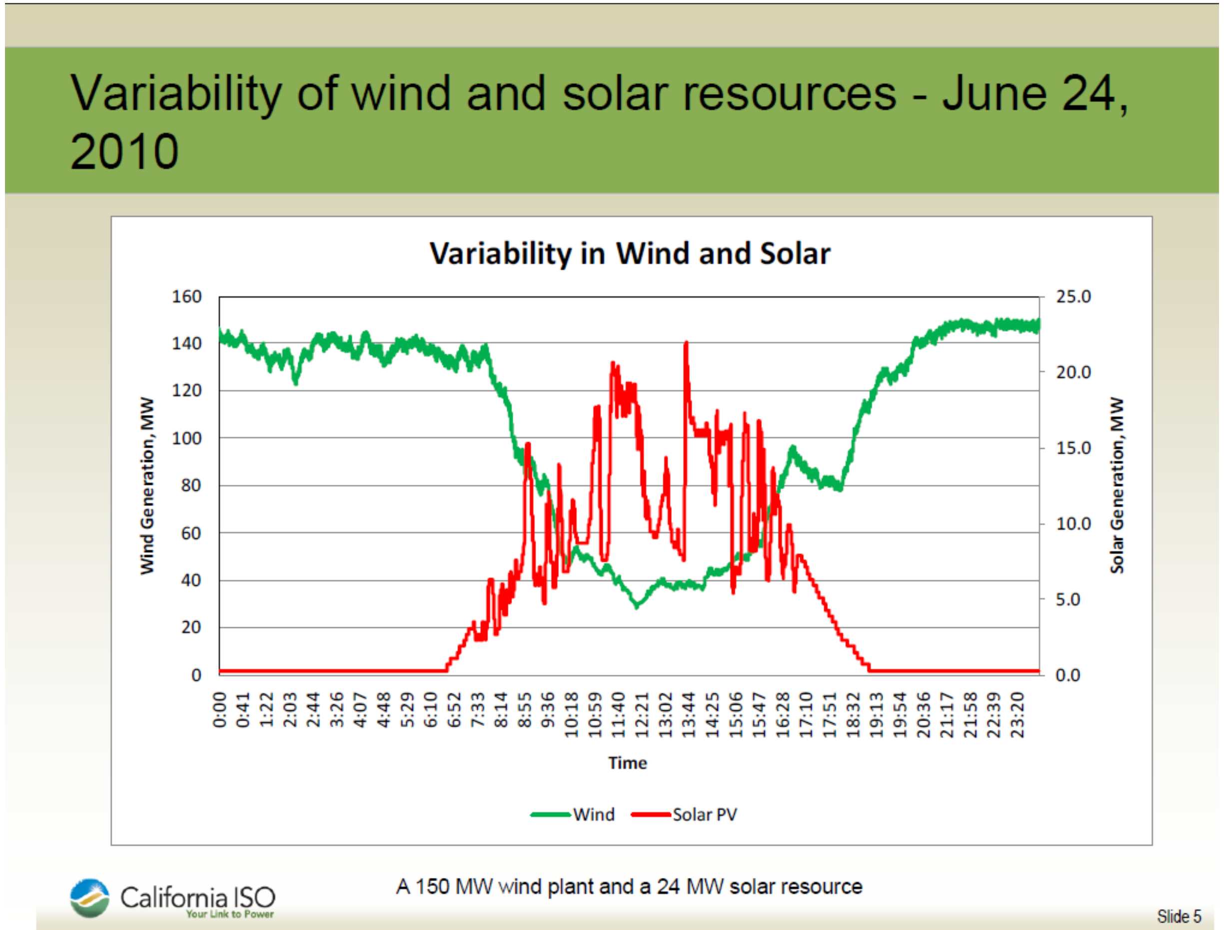
- Renewable energy integration
  - depresses electricity prices
  - requires flexibility due to
    - uncertainty,
    - variability,
    - non-controllability of output
- Demand is unresponsive
- Supply-demand must be balanced instantaneously
- Network constraints
  - Favorable sites for solar/wind power installations driven by weather conditions
  - But networks have been built over decades with conventional thermal power sources in mind





Challenges of renewable energy (II)





Context

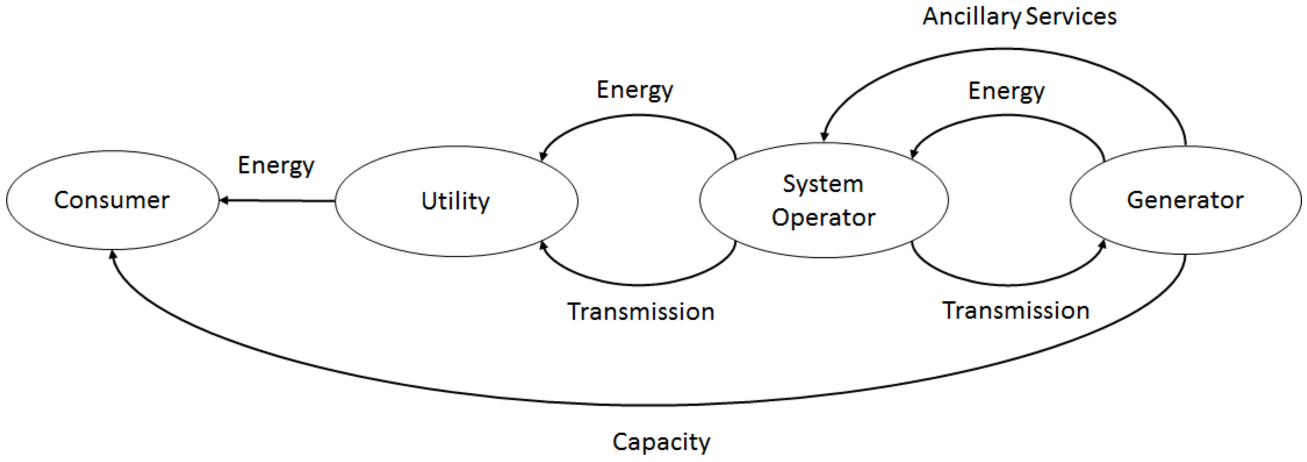
Renewable energy integration

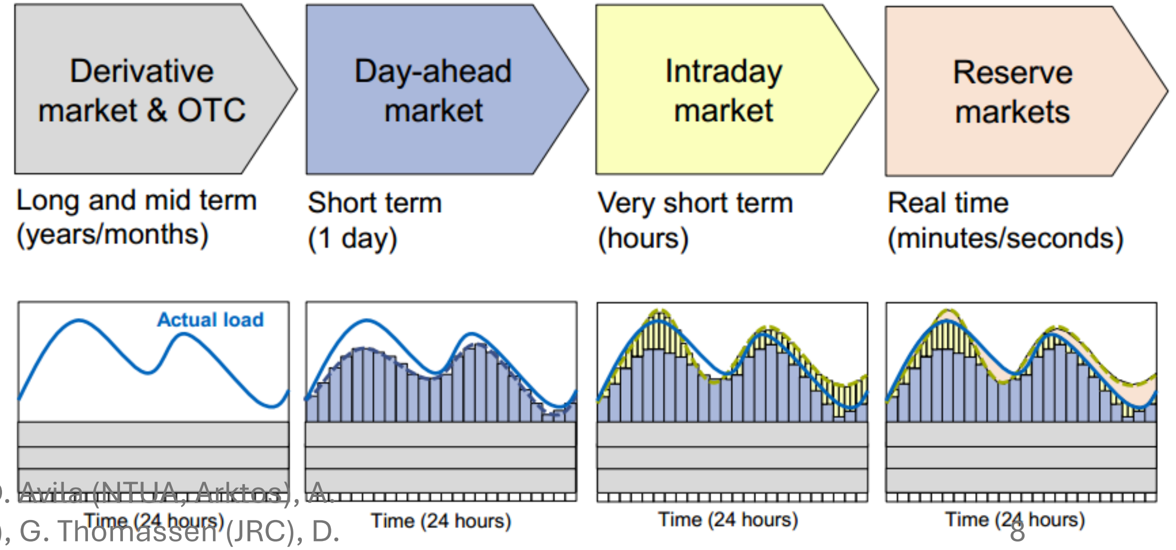
Capacity markets

Capacity expansion planning in the EU

Revenue streams in electricity markets

- Energy
  - Day-ahead ‘uniform price’ auction
- Reserve
  - Monthly procurement of reserve *capacity*
  - Real-time procurement of reserve *energy*
- Capacity
  - Auctioned annually in <u>some</u> markets
- Recent migration of value away from energy markets and into flexibility (reserves) and capacity (post energy crisis)





Renewable capacity auctions

- European market design is fundamentally problematic when it comes to steering investment in appropriate locations due to its reliance on **zonal pricing**
  - For instance, there is a unique electricity price in all of Germany
  - So an investor who is considering building a wind project in Germany is indifferent between the north and the south, despite the fact that the network is congested from north to south
- Corrections to the flawed zonal pricing design can be induced through **locational capacity auctions** [3]
- The goal of this project is to assess how much renewable capacity should be financed through capacity auctions in different parts of the European grid in order to meet the aforementioned aggressive renewable energy integration targets of the EU

Context

Renewable energy integration

Capacity markets

Capacity expansion planning in the EU

How we got here

- This work is a joint effort between NTUA and the European Commission Joint Research Center (JRC)
  - NTUA team: Daniel Avila, Alexandros Visas, Marilena Zampara, AP
  - JRC team: Georg Thomassen, David Pozo
- The EU has a standardized capacity expansion planning model on which this work builds
- This model is used for the **European Resource Adequacy Assessment**
  - Standardized capacity expansion model used for dimensioning capacity auctions, i.e. deciding how much capacity to finance in different Member States in which the EU has allowed for the introduction of State Aid through capacity auctions
  - The ERAA is managed/further developed by ENTSOE, with oversight of ACER
- In collaboration with ENTSOE, our team developed tailored decomposition algorithms for solving the stochastic version of the capacity expansion model used in ERAA [4]
- Subsequent work [5] focused on introducing endogenous reliability metrics to the model of [4]
- The current work focuses on introducing endogenous renewable energy targets to the model of [4]

Regulatory context for ERAA

L 158/54 EN Official Journal of the European Union 14.6.2019

REGULATION (EU) 2019/943 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL

of 5 June 2019

on the internal market for electricity

(recast)

(Text with EEA relevance)

- Regulation (EU) 2019/943, paragraph 43:

The ENTSO for Electricity should carry out a robust medium to long-term **European resource adequacy assessment** to provide an objective basis for the assessment of adequacy concerns.

The resource adequacy concern that **capacity mechanisms** address should be based on the European resource adequacy assessment. That assessment may be complemented by national assessments.

Regulatory context for ERAA

From adequacy assessments...

...to capacity requirements

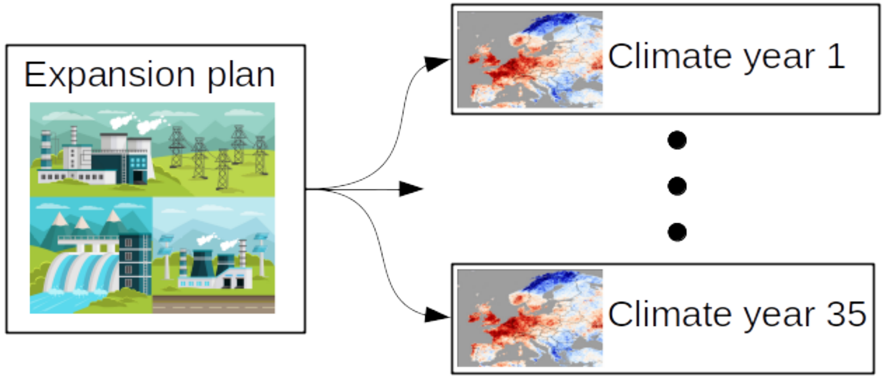
ACER
- Provides guidelines for the ERAA methodology
- Approves the ERAA methodology
- Approves the ERAA reports
- Provides guidelines for the determination of reliability criteria of Member States

ENTSOE
- Formulates the ERAA methodology
- Executes ERAA

National TSOs
- Are involved in ERAA
- Perform own adequacy assessments
- Justify deviations from ERAA
- Identify **gap** of capacities

Uncertainty in ERAA

- The **economic viability assessment (EVA)** is the component of ERAA which estimates how much capacity the market is expected to produce in the future under competitive conditions
- **The EVA aims at calculating an expansion plan** (expansion and retirement opportunities) of power plants for the entire European network
- However, **there is uncertainty** as we don’t know the precise climatic future conditions, but we have a set of possible future conditions
- **The expansion plan must behave optimally under uncertainty**, this leads us to a stochastic optimization problem



So far, the ability to solve a **stochastic EVA has been limited**

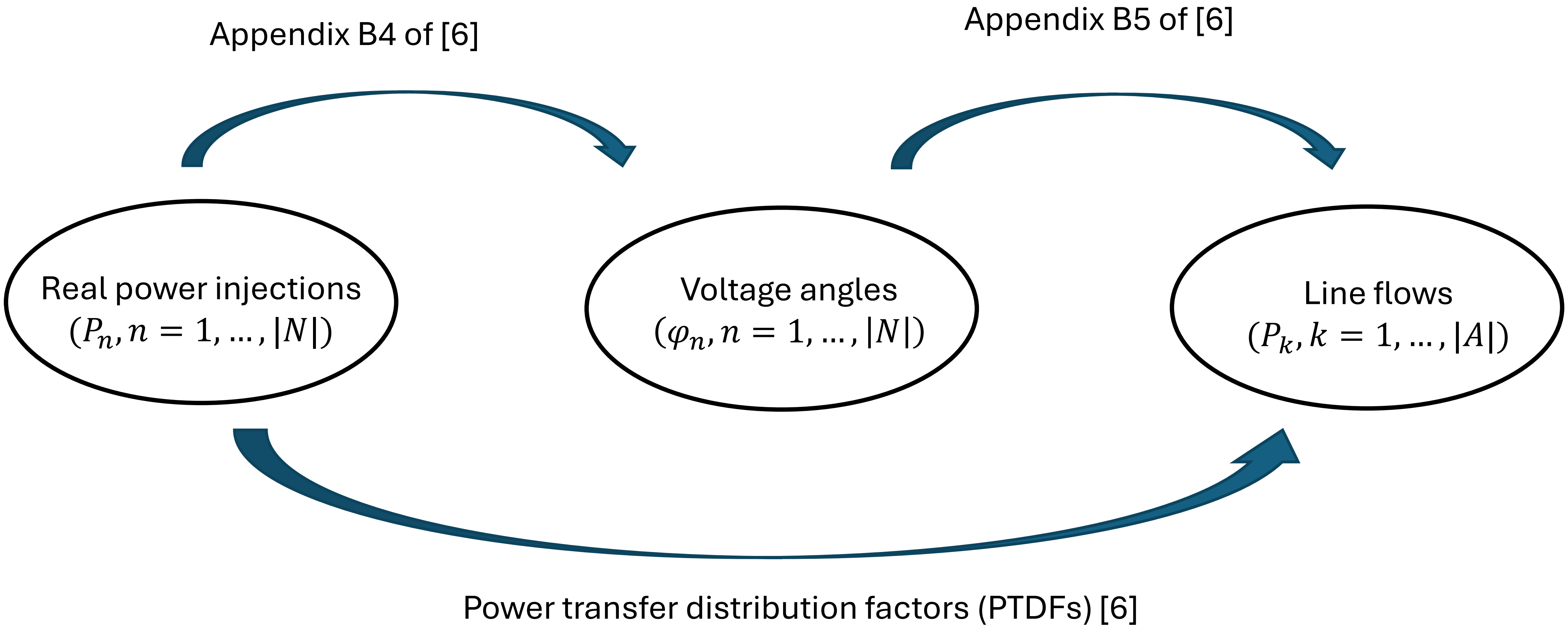
Power Flow

Kirchhoff’s laws

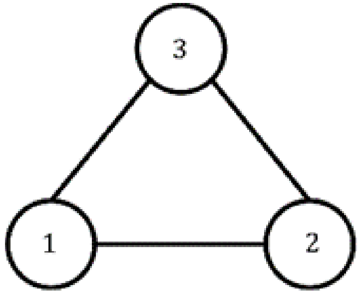
Kirchhoff’s current law: the total current flowing into a node equals the total current flowing out of a node

Kirchhoff’s voltage law: accumulated voltage change across any loop of an electrical circuit equals zero

Relations between injections, voltage angles and flows



Example: symmetric 3-node network



Denote X as the reactance of each line

Power transfer distribution factor $PTDF_{k,n}$ and maps how many MW of power flow from line k when shipping 1 MW of power from node n to the hib node

The PTDF matrix of the network when node 3 is the hub node is as follows:

$$PTDF = \begin{array}{c} \\ 1-2 \\ 2-3 \\ 1-3 \end{array} \begin{array}{c} \begin{array}{cc} 1 & 2 \end{array} \\ \begin{bmatrix} 1/3 & -1/3 \\ 1/3 & 2/3 \\ 2/3 & 1/3 \end{bmatrix} \end{array}$$

Capacity expansion model

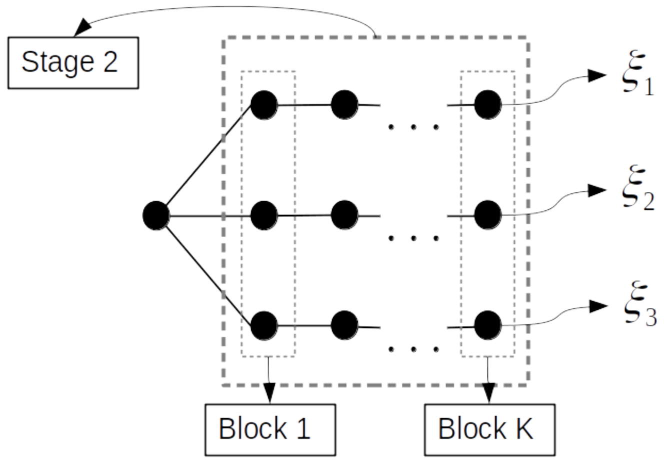
Modeling goal

Large-scale stochastic capacity expansion planning with endogenous target renewable energy constraints

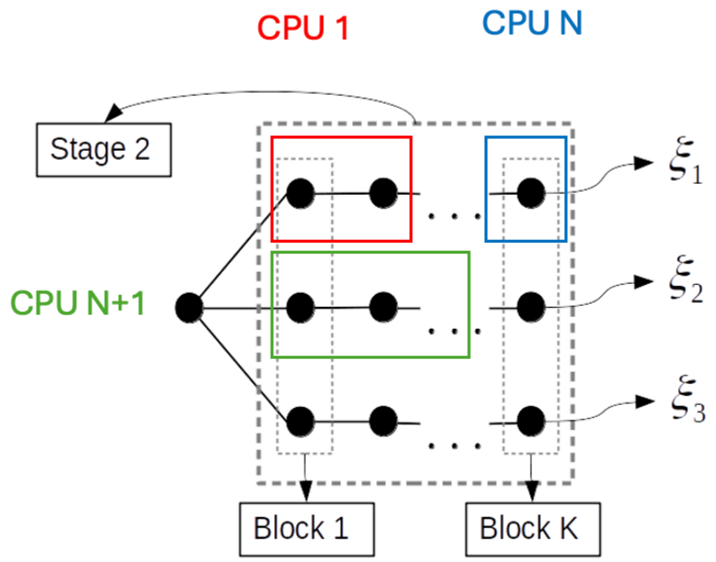
- European geographic coverage
- Hourly time resolution
- Representation of linearized power flow equations
- Representation of uncertainty
  - Several climatic scenarios
- Decisions on investments of renewables
  - Capacity of the asset
  - Location of the asset
- Volume of renewable production must meet desired renewable integration target

How to achieve it?

- Decomposition methods
  - Breakdown the overall problem into smaller tractable subproblems



- High performance computing infrastructure
  - Distribute the computational work in a large array of CPUs



Simple capacity expansion problem (CEP)

$$\min_{\substack{x,p,ls\geq 0\\ ne,f,\varphi}} \sum_{g\in G} I_g \cdot x_g + E_{\omega\in\Omega}\left[\sum_{g\in G}\sum_{t\in T} MC_g \cdot p_{t,g,\omega} + VOLL \sum_{t\in T} ls_{t,\omega}\right]$$

investment cost + operational cost

$$x_g \leq X_g \quad \forall g \in G$$

investment constraints

$$p_{g,t,\omega} \leq x_g \quad \forall g \in G, t \in T, \omega \in \Omega$$

power generation constraints

$$D_{t,\omega} - \sum_{g\in G} p_{t,g,\omega} - ls_{t,\omega} + ne_{t,\omega} = 0 \quad \forall t \in T, \omega \in \Omega$$

load balance

$$ne_{t,\omega} = \sum_{l=(n,\cdot)} f_{l,t,\omega} - \sum_{l=(\cdot,n)} f_{l,t,\omega} \quad \forall t \in T, \omega \in \Omega$$

$$f_{l,t,\omega} = B_k \cdot \left(\varphi_{n,t,\omega} - \varphi_{n,t,\omega}\right) \forall l = (n,m) \in K, t \in T, \omega \in \Omega$$

transmission network

$$F_l \leq f_{l,t,\omega} \leq F_l \quad \forall l = (n,m) \in K, t \in T, \omega \in \Omega$$

Simple capacity expansion problem (CEP)

$$\min_{\substack{x,p,ls\geq 0\\ ne,f,\varphi}} \sum_{g\in G} I_g \cdot x_g + E_{\omega\in\Omega}\left[\sum_{g\in G}\sum_{t\in T} MC_g \cdot p_{t,g,\omega} + VOLL \sum_{t\in T} ls_{t,\omega}\right]$$

$$x_g \leq X_g \quad \forall g \in G$$

$$p_{g,t,\omega} \leq x_g \quad \forall g \in G, t \in T, \omega \in \Omega$$

$$D_{t,\omega} - \sum_{g\in G} p_{t,g,\omega} - ls_{t,\omega} + ne_{t,\omega} = 0 \quad \forall t \in T, \omega \in \Omega$$

$$ne_{t,\omega} = \sum_{l=(n,\cdot)} f_{l,t,\omega} - \sum_{l=(\cdot,n)} f_{l,t,\omega} \quad \forall t \in T, \omega \in \Omega$$

$$f_{l,t,\omega} = B_k \cdot \left(\varphi_{n,t,\omega} - \varphi_{n,t,\omega}\right) \forall l = (n,m) \in K, t \in T, \omega \in \Omega$$

$$F_l \leq f_{l,t,\omega} \leq F_l \quad \forall l = (n,m) \in K, t \in T, \omega \in \Omega$$

- Two-stage formulation (**first stage** and then **second stage** depending on first stage variables)
- We can directly implement decomposition methods

CEP with target volume constraint

$$\min_{\substack{x,p,ls \geq 0 \\ ne,f,\varphi}} \sum_{g \in G} I_g \cdot x_g + E_{\omega \in \Omega}\left[\sum_{g \in G} \sum_{t \in T} MC_g \cdot p_{t,g,\omega} + VOLL \sum_{t \in T} ls_{t,\omega}\right]$$

$$x_g \leq X_g \quad \forall g \in G$$

$$p_{g,t,\omega} \leq x_g \quad \forall g \in G, t \in T, \omega \in \Omega$$

$$D_{t,\omega} - \sum_{g \in G} p_{t,g,\omega} - ls_{t,\omega} + ne_{t,\omega} = 0 \quad \forall t \in T, \omega \in \Omega$$

$$ne_{t,\omega} = \sum_{l=(n,\cdot)} f_{l,t,\omega} - \sum_{l=(\cdot,n)} f_{l,t,\omega} \quad \forall t \in T, \omega \in \Omega$$

$$f_{l,t,\omega} = B_k \cdot (\varphi_{n,t,\omega} - \varphi_{n,t,\omega}) \; \forall l = (n,m) \in K, t \in T, \omega \in \Omega$$

$$F_l \leq f_{l,t,\omega} \leq F_l \quad \forall l = (n,m) \in K, t \in T, \omega \in \Omega$$

$$E_{\omega \in \Omega}\left[\sum_{g \in G} \sum_{t \in T} p_{t,g,\omega}\right] = \mathrm{TV}$$

- The **target volume constraint** bundles scenario-specific variables
- We cannot directly implement decomposition

Algorithmics

Decoupling the target volume constraint

Scenario decomposition

Regularization

Temporal decomposition

Parallel computing implementation

Target volume decoupling

- We resort on the notion of budget variables [7]
  - A. Jacobson, F. Pecci, N. Sepulveda, Q. Xu, and J. Jenkins, “A computationally efficient Benders decomposition for energy systems planning problems with detailed operations and time-coupling constraints,” INFORMS Journal on Optimization, vol. 6, 2023.
- A “budget” variable is introduced for each scenario
  - This budget variable is the amount of production volume for each scenario
  - The budget variables are decided during the first stage: thus, the target volume constraint becomes a first-stage constraint

Mathematical description

$$\min_{\substack{x,p,ls,q\geq 0\\ ne,f,\varphi}} \sum_{g\in G} I_g \cdot x_g + E_{\omega\in\Omega}\left[\sum_{g\in G}\sum_{t\in T} MC_g \cdot p_{t,g,\omega} + VOLL \sum_{t\in T} ls_{t,\omega}\right]$$

$$x_g \leq X_g \quad \forall g \in G$$

$$p_{g,t,\omega} \leq x_g \quad \forall g \in G, t \in T, \omega \in \Omega$$

$$D_{t,\omega} - \textstyle\sum_{g\in G} p_{t,g,\omega} - ls_{t,\omega} + ne_{t,\omega} = 0 \quad \forall t \in T, \omega \in \Omega$$

$$ne_{t,\omega} = \textstyle\sum_{l=(n,\cdot)} f_{l,t,\omega} - \sum_{l=(\cdot,n)} f_{l,t,\omega} \quad \forall t \in T, \omega \in \Omega$$

$$f_{l,t,\omega} = B_k \cdot (\varphi_{n,t,\omega} - \varphi_{n,t,\omega})\ \forall l = (n,m) \in K, t \in T, \omega \in \Omega$$

$$F_l \leq f_{l,t,\omega} \leq F_l \quad \forall l = (n,m) \in K, t \in T, \omega \in \Omega$$

$$E_{\omega\in\Omega}\left[\sum_{g\in G}\sum_{t\in T} p_{t,g,\omega}\right] = \mathrm{TV}$$

Instead of this constraint, we introduce a budget q_ω so that $\sum_{g\in G}\sum_{t\in T} p_{t,g,\omega} = q_\omega$

Mathematical description

- The budgets become first-stage decisions
- Each scenario respects the decided budget
- We are left with a standard two-stage problem

$$\min_{\substack{x,p,ls,q\geq 0\\ ne,f,\varphi}} \sum_{g\in G} I_g \cdot x_g + E_{\omega\in\Omega}\left[\sum_{g\in G}\sum_{t\in T} MC_g \cdot p_{t,g,\omega} + VOLL \sum_{t\in T} ls_{t,\omega}\right]$$

$$x_g \leq X_g \quad \forall g \in G$$

$$E_{\omega\in\Omega}[q_\omega] = \text{TV}$$

$$p_{g,t,\omega} \leq x_g \quad \forall g \in G, t \in T, \omega \in \Omega$$

$$D_{t,\omega} - \sum_{g\in G} p_{t,g,\omega} - ls_{t,\omega} + ne_{t,\omega} = 0 \quad \forall t \in T, \omega \in \Omega$$

$$ne_{t,\omega} = \sum_{l=(n,\cdot)} f_{l,t,\omega} - \sum_{l=(\cdot,n)} f_{l,t,\omega} \quad \forall t \in T, \omega \in \Omega$$

$$f_{l,t,\omega} = B_k \cdot (\varphi_{n,t,\omega} - \varphi_{n,t,\omega}) \; \forall l = (n,m) \in K, t \in T, \omega \in \Omega$$

$$F_l \leq f_{l,t,\omega} \leq F_l \quad \forall l = (n,m) \in K, t \in T, \omega \in \Omega$$

$$\sum_{g\in G}\sum_{t\in T} p_{t,g,\omega} = q_\omega \; \forall \omega \in \Omega$$

Algorithmics

Decoupling the target volume constraint

Scenario decomposition

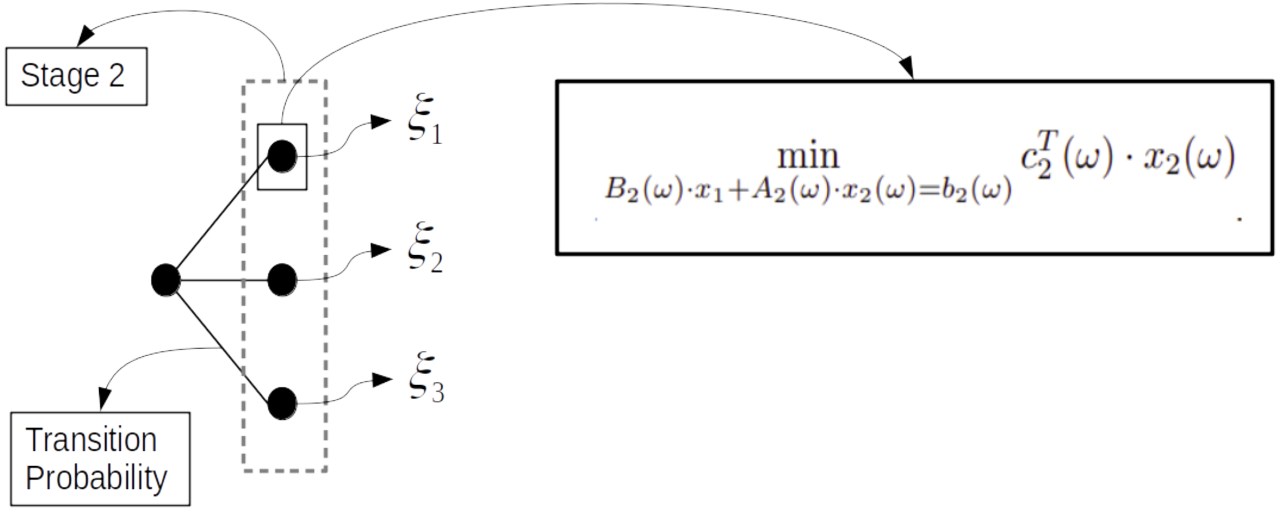
Regularization

Temporal decomposition

Parallel computing implementation

Scenario decomposition

- We fall back to the standard two-stage problem structure
  - The problem is decomposed into a master and slave subproblems
- The master problem decides the investment and budget decisions
- For each scenario there is a slave subproblem
  - Corresponds to the economic dispatch ***given*** the scenario and first-stage decisions



Mathematical description

We follow the standard Bender's decomposition approach

$$\min_{x,q,\theta\geq 0} \sum_{g\in G} I_g \cdot x_g + E[\theta_\omega]$$

$$x_g \leq X_g \quad \forall g \in G$$

$$E[q_\omega] = TV$$

$$(\theta_\omega, x_g, q_\omega) \in Opt(\omega)$$

The master problem has an approximation of the value function

The value function is approximated through optimality cuts

$$\min_{\substack{p,ls\geq 0\\ ne.f.\varphi}} \sum_{g\in G}\sum_{t\in T} MC_g \cdot p_{t,g,\omega} + VOLL \sum_{t\in T} ls_{t,\omega}$$

$$p_{g,t,\omega} \leq x_g \quad \forall g \in G, t \in T$$

$$D_{t,\omega} - \sum_{g\in G} p_{t,g,\omega} - ls_{t,\omega} + ne_{t,\omega} = 0 \quad \forall t \in T$$

$$ne_{t,\omega} = \sum_{l=(n,\cdot)} f_{l,t,\omega} - \sum_{l=(\cdot,n)} f_{l,t,\omega} \quad \forall t \in T$$

$$f_{l,t,\omega} = B_k \cdot \left(\varphi_{n,t,\omega} - \varphi_{n,t,\omega}\right) \forall l = (n,m) \in K, t \in T$$

$$F_l \leq f_{l,t,\omega} \leq F_l \quad \forall l = (n,m) \in K, t \in T$$

$$\sum_{g\in G}\sum_{t\in T} p_{t,g,\omega} = q_\omega$$

For each scenario, we have a slave problem that is an economic dispatch problem, with a budget target

Mathematical description

$$\min_{x,q,\theta\geq 0} \sum_{g\in G} I_g \cdot x_g + E[\theta_\omega]$$

$$x_g \leq X_g \quad \forall g \in G$$

$$E[q_\omega] = TV$$

$$(\theta_\omega, x_g, q_\omega) \in Opt(\omega)$$

$$\min_{\substack{p,ls,q\geq 0\\ ne,f,\varphi}} \sum_{g\in G}\sum_{t\in T} MC_g \cdot p_{t,g,\omega} + VOLL \sum_{t\in T} ls_{t,\omega}$$

$$p_{g,t,\omega} \leq x_g \quad \forall g \in G, t \in T$$

$$D_{t,\omega} - \sum_{g\in G} p_{t,g,\omega} - ls_{t,\omega} + ne_{t,\omega} = 0 \quad \forall t \in T$$

$$ne_{t,\omega} = \sum_{l=(n,\cdot)} f_{l,t,\omega} - \sum_{l=(\cdot,n)} f_{l,t,\omega} \quad \forall t \in T$$

$$f_{l,t,\omega} = B_k \cdot (\varphi_{n,t,\omega} - \varphi_{n,t,\omega})\ \forall l = (n,m) \in K, t \in T$$

$$F_l \leq f_{l,t,\omega} \leq F_l \quad \forall l = (n,m) \in K, t \in T$$

$$\sum_{g\in G}\sum_{t\in T} p_{t,g,\omega} = q_\omega$$

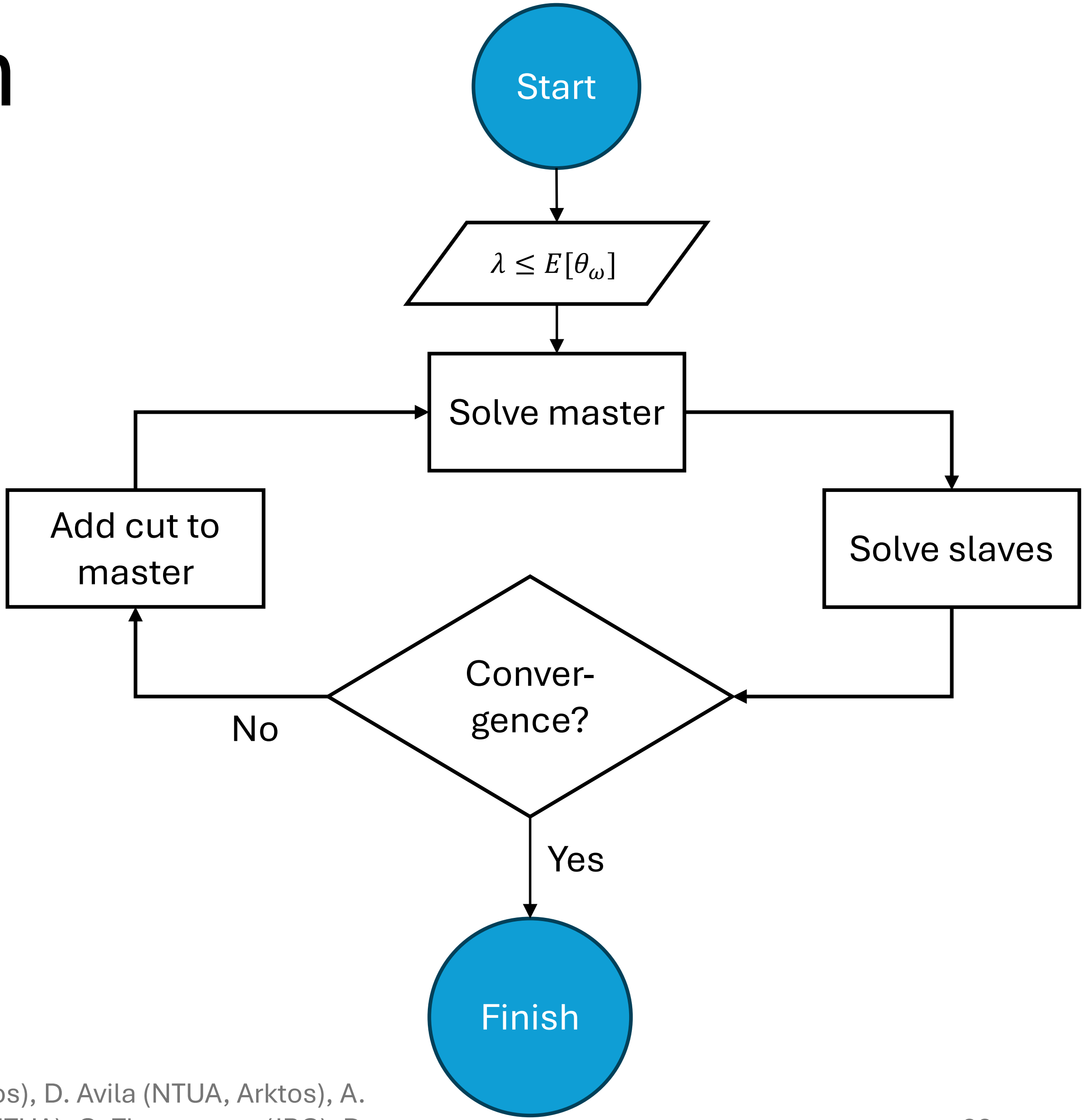
We have a well-defined optimality gap

The master problem objective value serves as a **lower bound**

The average economic dispatch cost + investment cost serves as an **upper bound**

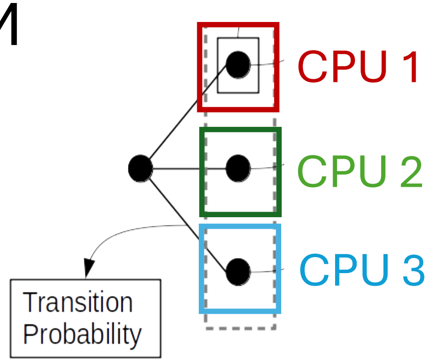
Algorithmic description

1. Provide initial lower bound value for $E[\theta_\omega]$
2. Solve master problem
3. Solve slave subproblems
4. Add cut to master problem



Limitations

- In practice, convergence can be slow
  - Initial iterations are not particularly intelligent, because value functions are bad approximations of second-stage costs
  - The early iterates tend to waste time exploring the corners of the feasible space
- Subproblems are large-scale LPs
  - Iterations are costly
  - A full-EU system at nodal level becomes restrictive due to RAM requirements
- Limitations of high-performance computing
  - The number of CPUs is limited by the number of scenarios



How to deal with these limitations?

- In practice, convergence can be slow
  - **We can take advantage of regularization techniques**
- Subproblems are large-scale LPs
  - **We develop a further decomposition of the subproblems**
- Limitations of high-performance computing
  - **We develop a distributed computing implementation that is no longer limited by the number of scenarios**

Algorithmics

Decoupling the target volume constraint

Scenario decomposition

Regularization

Temporal decomposition

Parallel computing implementation

Regularization

- The idea of regularization is to penalize large deviations from the best iterate found so far
  - The advantage is that we avoid bouncing between corners of the feasible space

- It allows to “warm-start” the algorithm at a known “reasonable” solution
  - It allows narrowing down the search and avoid building value function approximations at regions far from optimality
  - We propose to use the average solution obtained with the wait-and-see solution as a “reasonable” starting point

Mathematical description

The regularization is introduced at the level of the master problem

$$\min_{x,q,\theta\geq 0} \sum_{g\in G} I_g \cdot x_g + E[\theta_\omega] + \boxed{\rho_1 \cdot \left|\left|x - x^v\right|\right|^2 + \rho_2 \cdot \left|\left|q - q^v\right|\right|^2}$$

$$x_g \leq X_g \quad \forall g \in G$$

$$E[q_\omega] = TV$$

$$\left(\theta_\omega, x_g, q_\omega\right) \in Opt(\omega)$$

Mathematical description

The regularization is introduced at the level of the master problem

$$\min_{x,q,\theta\geq 0} \sum_{g\in G} I_g \cdot x_g + E[\theta_\omega] + \rho_1 \cdot ||x - x^v||^2 + \rho_2 \cdot ||q - q^v||^2$$

$$x_g \leq X_g \quad \forall g \in G$$

$$E[q_\omega] = TV$$

$$(\theta_\omega, x_g, q_\omega) \in Opt(\omega)$$

The problem becomes a quadratic convex problem

- The master problem is small, thus remains tractable
- There are techniques to improve the performance of the master, e.g. cut selection techniques

Mathematical description

$$\min_{x,q,\theta\geq 0} \sum_{g\in G} I_g \cdot x_g + E[\theta_\omega] + \rho_1 \cdot ||x - x^v||^2 + \rho_2 \cdot ||q - q^v||^2$$

$$x_g \leq X_g \quad \forall g \in G$$

$$E[q_\omega] = TV$$

$$(\theta_\omega, x_g, q_\omega) \in Opt(\omega)$$

The regularization is introduced at the level of the master problem

The problem becomes a quadratic convex problem

- The master problem is small thus remains tractable
- There are techniques to improve the performance of the master, e,g, cut selection techniques

The ρ parameters have to be adjusted

- We employ adaptive regularization schemes

Mathematical description

The regularization is introduced at the level of the master problem

$$\min_{x,q,\theta\geq 0} \sum_{g\in G} I_g \cdot x_g + E[\theta_\omega] + \rho_1 \cdot \left\|x - x^v\right\|^2 + \rho_2 \cdot \left\|q - q^v\right\|^2$$

$$x_g \leq X_g \quad \forall g \in G$$

$$E[q_\omega] = TV$$

$$(\theta_\omega, x_g, q_\omega) \in Opt(\omega)$$

The problem becomes a quadratic convex problem

- The master problem is small thus remains tractable
- There are techniques to improve the performance of the master, e.g. cut selection techniques

The ρ parameters have to be adjusted

- We employ adaptive regularization schemes

The x^v, q^v are the best solution estimates

- We employ the average wait-and-see solution as initial solution

Mathematical description

$$\min_{x,q,\theta\geq 0} \sum_{g\in G} I_g \cdot x_g + E[\theta_\omega] + \boxed{\rho_1 \cdot ||x - x^v||^2 + \rho_2 \cdot ||q - q^v||^2}$$

$$x_g \leq X_g \quad \forall g \in G$$

$$E[q_\omega] = TV$$

$$(\theta_\omega, x_g, q_\omega) \in Opt(\omega)$$

The regularization is introduced at the level of the master problem

The problem becomes a quadratic convex problem

- The master problem is small thus remains tractable
- There are techniques to improve the performance of the master, e.g. cut selection techniques

The ρ parameters have to be adjusted

- We employ adaptive regularization schemes

The regularized master problem no longer produces a lower bound

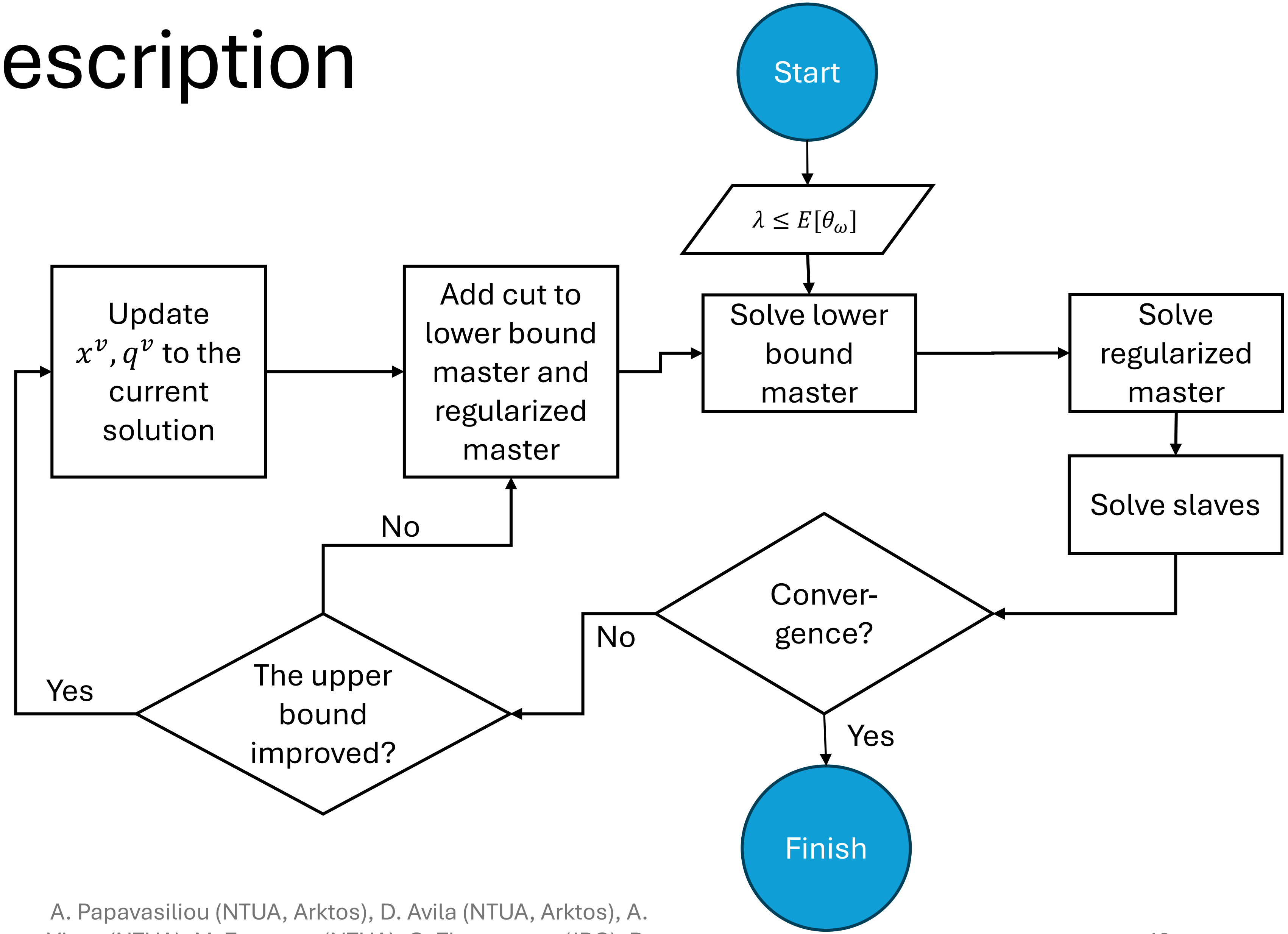
- We introduce an auxiliary non-regularized master problem for computing a lower bound

The x^v, q^v are the best solution estimates

- We employ the average wait-and-see solution as initial solution

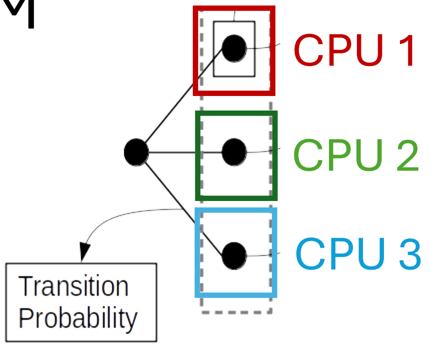
Algorithmic description

1. Provide initial lower bound value for $E[\theta_\omega]$, and initial x^v, q^v estimates
2. Solve lower bound master problem
3. Solve regularized master problem
4. Solve slave subproblems
5. Update master problems



Limitations

- In practice the convergence can be slow
  - Initial iterations are not particularly intelligent, because value functions are bad approximations of second-stage costs
  - The iterations tend to waste time exploring the corners of the feasible space
- Subproblems are large-scale LPs
  - Iterations are costly
  - A full-EU system at nodal level becomes restrictive due to RAM requirements
- Limitations of high-performance computing
  - The number of CPUs is limited by the number of scenarios



Algorithmics

Decoupling target volume constraint

Scenario decomposition

Regularization

Temporal decomposition

Parallel computing implementation

Temporal decomposition

- We focus on an economic dispatch subproblem for a given scenario
  - We decompose the economic dispatch subproblem into an array of subproblems
  - The idea is that throughout iterations the solution of the array of subproblems converges to the economic dispatch solution
- The problematic constraints are those that relate to inter-temporal behavior
  - Long-duration storage constraints

Storage modeling

$$\min_{\substack{p,ls,ph,e\geq 0\\ ne,f,\varphi}} \sum_{g\in G}\sum_{t\in T} MC_g \cdot p_{t,g,\omega} + VOLL \sum_{t\in T} ls_{t,\omega}$$

$$p_{g,t,\omega} \leq x_g \quad \forall g \in G, t \in T$$

$$D_{t,\omega} - \sum_{g\in G} p_{t,g,\omega} - ls_{t,\omega} + ne_{t,\omega} \boxed{- \sum_{g\in Gh} ph_{t,g,\omega} = 0} \quad \forall t \in T$$

$$ne_{t,\omega} = \sum_{l=(n,\cdot)} f_{l,t,\omega} - \sum_{l=(\cdot,n)} f_{l,t,\omega} \quad \forall t \in T$$

$$f_{l,t,\omega} = B_k \cdot \left(\varphi_{n,t,\omega} - \varphi_{n,t,\omega}\right) \forall l = (n,m) \in K, t \in T$$

$$F_l \leq f_{l,t,\omega} \leq F_l \quad \forall l = (n,m) \in K, t \in T$$

$$\sum_{g\in G}\sum_{t\in T} p_{t,g,\omega} = q_\omega$$

$$\boxed{\begin{aligned} & e_{g,t,\omega} = e_{g,t-1,\omega} - ph_{g,t,\omega} + A_{g,t,\omega} \quad \forall g \in Gh, t \in T \\ & e_{g,t,\omega} \leq E_{g,t,\omega} \quad \forall g \in Gh, t \in T \end{aligned}}$$

The basic storage model includes:

The energy from storage in the load balance constraint

The energy balance constraints from storage

These are time-coupling

Economic dispatch subproblem

$$\min_{\substack{p,ls,ph,e\geq 0\\ ne,f,\varphi}} \sum_{g\in G} \boxed{\sum_{t\in T_k}} MC_g \cdot p_{t,g,\omega} + VOLL \boxed{\sum_{t\in T_k}} ls_{t,\omega}$$

The time horizon is subdivided into blocks
A subproblem is built for every block

$$p_{g,t,\omega} \leq x_g \quad \forall g \in G, t \in T_k$$

$$D_{t,\omega} - \textstyle\sum_{g\in G} p_{t,g,\omega} - ls_{t,\omega} + ne_{t,\omega} - \sum_{g\in Gh} ph_{t,g,\omega} = 0 \quad \forall t \in T_k$$

$$ne_{t,\omega} = \textstyle\sum_{l=(n,\cdot)} f_{l,t,\omega} - \sum_{l=(\cdot,n)} f_{l,t,\omega} \quad \forall t \in T_k$$

$$f_{l,t,\omega} = B_k \cdot \left(\varphi_{n,t,\omega} - \varphi_{n,t,\omega}\right) \forall l = (n,m) \in K, t \in T_k$$

$$F_l \leq f_{l,t,\omega} \leq F_l \quad \forall l = (n,m) \in K, t \in T_k$$

$$\sum_{g\in G} \sum_{t\in T_k} p_{t,g,\omega} = q_{k,\omega}$$

$$e_{g,t,\omega} = e_{g,t-1,\omega} - ph_{g,t,\omega} + A_{g,t,\omega} \quad \forall g \in Gh, t \in T_k$$

$$e_{g,t,\omega} \leq E_{g,t,\omega} \quad \forall g \in Gh, t \in T_k$$

Economic dispatch subproblem

$$\min_{\substack{p,ls,ph,e\geq 0\\ ne,f,\varphi}} \sum_{g\in G}\sum_{\boxed{t\in T_k}} MC_g \cdot p_{t,g,\omega} + VOLL \sum_{\boxed{t\in T_k}} ls_{t,\omega}$$

$$p_{g,t,\omega} \leq x_g \quad \forall g\in G, t\in T_k$$

$$D_{t,\omega} - \sum_{g\in G} p_{t,g,\omega} - ls_{t,\omega} + ne_{t,\omega} - \sum_{g\in Gh} ph_{t,g,\omega} = 0 \quad \forall t\in T_k$$

$$ne_{t,\omega} = \sum_{l=(n,\cdot)} f_{l,t,\omega} - \sum_{l=(\cdot,n)} f_{l,t,\omega} \quad \forall t\in T_k$$

$$f_{l,t,\omega} = B_k \cdot \left(\varphi_{n,t,\omega} - \varphi_{n,t,\omega}\right) \forall l=(n,m)\in K, t\in T_k$$

$$F_l \leq f_{l,t,\omega} \leq F_l \quad \forall l=(n,m)\in K, t\in T_k$$

$$\sum_{g\in G}\sum_{t\in T_k} p_{t,g,\omega} = q_{k,\omega}$$

$$e_{g,t,\omega} = e_{g,t-1,\omega} - ph_{g,t,\omega} + A_{g,t,\omega} \quad \forall g\in Gh, t\in T_k$$

$$e_{g,t,\omega} \leq E_{g,t,\omega} \quad \forall g\in Gh, t\in T_k$$

The time horizon is subdivided in blocks

A subproblem is built for every block

We note that a budget variable is introduced for every block

Note that at the boundary the subproblem requires an input storage decision

How to deal with the boundary?

$$\min_{\substack{p,ls,ph,e\geq 0\\ ne,f,\varphi}} \sum_{g\in G}\sum_{t\in T_k} MC_g \cdot p_{t,g,\omega} + VOLL \sum_{t\in T_k} ls_{t,\omega}$$

$$p_{g,t,\omega} \leq x_g \quad \forall g \in G, t \in T_k$$

$$D_{t,\omega} - \sum_{g\in G} p_{t,g,\omega} - ls_{t,\omega} + ne_{t,\omega} - \sum_{g\in Gh} ph_{t,g,\omega} = 0 \quad \forall t \in T_k$$

$$ne_{t,\omega} = \sum_{l=(n,\cdot)} f_{l,t,\omega} - \sum_{l=(\cdot,n)} f_{l,t,\omega} \quad \forall t \in T_k$$

$$f_{l,t,\omega} = B_k \cdot \left(\varphi_{n,t,\omega} - \varphi_{n,t,\omega}\right) \forall l = (n,m) \in K, t \in T_k$$

$$F_l \leq f_{l,t,\omega} \leq F_l \quad \forall l = (n,m) \in K, t \in T_k$$

$$\sum_{g\in G}\sum_{t\in T_k} p_{t,g,\omega} = q_{k,\omega}$$

$$e_{g,t,\omega} = e_{g,t-1,\omega} - ph_{g,t,\omega} + A_{g,t,\omega} \quad \forall g \in Gh, t \in T_k$$

$$e_{g,t,\omega} \leq E_{g,t,\omega} \quad \forall g \in Gh, t \in T_k$$

$$e_{g,t_1-1,\omega} = \hat{e}_{g,t_1-1,\omega} \quad \forall g \in Gh$$

$$e_{g,t_2,\omega} = \hat{e}_{g,t_2,\omega} \quad \forall g \in Gh$$

The subproblem receives as an input storage decisions for the boundary

How to deal with the boundary?

$$\min_{\substack{p,ls,ph,e\geq 0\\ ne,f,\varphi}} \sum_{g\in G}\sum_{t\in T_k} MC_g \cdot p_{t,g,\omega} + VOLL \sum_{t\in T_k} ls_{t,\omega}$$

$$p_{g,t,\omega} \leq x_g \quad \forall g \in G, t \in T_k$$

$$D_{t,\omega} - \sum_{g\in G} p_{t,g,\omega} - ls_{t,\omega} + ne_{t,\omega} - \sum_{g\in Gh} ph_{t,g,\omega} = 0 \quad \forall t \in T_k$$

$$ne_{t,\omega} = \sum_{l=(n,\cdot)} f_{l,t,\omega} - \sum_{l=(\cdot,n)} f_{l,t,\omega} \quad \forall t \in T_k$$

$$f_{l,t,\omega} = B_k \cdot \left(\varphi_{n,t,\omega} - \varphi_{n,t,\omega}\right) \forall l = (n,m) \in K, t \in T_k$$

$$F_l \leq f_{l,t,\omega} \leq F_l \quad \forall l = (n,m) \in K, t \in T_k$$

$$\sum_{g\in G}\sum_{t\in T_k} p_{t,g,\omega} = q_{k,\omega}$$

$$e_{g,t,\omega} = e_{g,t-1,\omega} - ph_{g,t,\omega} + A_{g,t,\omega} \quad \forall g \in Gh, t \in T_k$$

$$e_{g,t,\omega} \leq E_{g,t,\omega} \quad \forall g \in Gh, t \in T_k$$

$$e_{g,t_1-1,\omega} = \hat{e}_{g,t_1-1,\omega} \quad \forall g \in Gh$$

$$e_{g,t_2,\omega} = \hat{e}_{g,t_2,\omega} \quad \forall g \in Gh$$

The subproblem receives as an input storage decisions for the boundary

Following the same idea as for “budget” volumes, these input decisions can be thought as “budget” storages

How to deal with the boundary?

$$\min_{\substack{p,ls,ph,e\geq 0\\ ne,f,\varphi}} \sum_{g\in G}\sum_{t\in T_k} MC_g \cdot p_{t,g,\omega} + VOLL \sum_{t\in T_k} ls_{t,\omega}$$

$$p_{g,t,\omega} \leq x_g \quad \forall g \in G, t \in T_k$$

$$D_{t,\omega} - \sum_{g\in G} p_{t,g,\omega} - ls_{t,\omega} + ne_{t,\omega} - \sum_{g\in Gh} ph_{t,g,\omega} = 0 \quad \forall t \in T_k$$

$$ne_{t,\omega} = \sum_{l=(n,\cdot)} f_{l,t,\omega} - \sum_{l=(\cdot,n)} f_{l,t,\omega} \quad \forall t \in T_k$$

$$f_{l,t,\omega} = B_k \cdot \left(\varphi_{n,t,\omega} - \varphi_{n,t,\omega}\right) \forall l = (n,m) \in K, t \in T_k$$

$$F_l \leq f_{l,t,\omega} \leq F_l \quad \forall l = (n,m) \in K, t \in T_k$$

$$\sum_{g\in G}\sum_{t\in T_k} p_{t,g,\omega} = q_{k,\omega}$$

$$e_{g,t,\omega} = e_{g,t-1,\omega} - ph_{g,t,\omega} + A_{g,t,\omega} \quad \forall g \in Gh, t \in T_k$$

$$e_{g,t,\omega} \leq E_{g,t,\omega} \quad \forall g \in Gh, t \in T_k$$

$$\boxed{\begin{aligned} e_{g,t_1-1,\omega} &= \hat{e}_{g,t_1-1,\omega} \quad \forall g \in Gh \\ e_{g,t_2,\omega} &= \hat{e}_{g,t_2,\omega} \quad \forall g \in Gh \end{aligned}}$$

The subproblem receives as an input storage decisions for the boundary

Following the same idea as for “budget” volumes, these input decisions can be thought as “budget” storages

These “budget” storages become first-stage decisions

The master problem handles these decisions

Infeasibilities

$$\min_{\substack{p,ls,ph,e\geq 0\\ ne,f,\varphi\\ pe^{+/-}\geq 0}} \sum_{g\in G}\sum_{t\in T_k} MC_g \cdot p_{t,g,\omega} + VOLL \sum_{t\in T_k} ls_{t,\omega} + PEN \sum_{g\in Gh} (pe^+_{g,t_1-1,\omega} + pe^-_{g,t_1-1,\omega} + pe^+_{g,t_2,\omega} + pe^-_{g,t_2,\omega})$$

$$p_{g,t,\omega} \leq x_g \quad \forall g \in G, t \in T_k$$

$$D_{t,\omega} - \sum_{g\in G} p_{t,g,\omega} - ls_{t,\omega} + ne_{t,\omega} - \sum_{g\in Gh} ph_{t,g,\omega} = 0 \quad \forall t \in T_k$$

$$ne_{t,\omega} = \sum_{l=(n,\cdot)} f_{l,t,\omega} - \sum_{l=(\cdot,n)} f_{l,t,\omega} \quad \forall t \in T_k$$

$$f_{l,t,\omega} = B_k \cdot \left(\theta_{n,t,\omega} - \theta_{n,t,\omega}\right) \forall l = (n,m) \in K, t \in T_k$$

$$F_l \leq f_{l,t,\omega} \leq F_l \quad \forall l = (n,m) \in K, t \in T_k$$

$$\sum_{g\in G}\sum_{t\in T_k} p_{t,g,\omega} = q_{k,\omega}$$

$$e_{g,t,\omega} = e_{g,t-1,\omega} - ph_{g,t,\omega} + A_{g,t,\omega} \quad \forall g \in Gh, t \in T_k$$

$$e_{g,t,\omega} \leq E_{g,t,\omega} \quad \forall g \in Gh, t \in T_k$$

$$e_{g,t_1-1,\omega} = \hat{e}_{g,t_1-1,\omega} + pe^+_{g,t_1-1,\omega} - pe^-_{g,t_1-1,\omega} \ \forall g \in Gh$$

$$e_{g,t_2,\omega} = \hat{e}_{g,t_2,\omega} + pe^+_{g,t_2,\omega} - pe^-_{g,t_2,\omega} \ \forall g \in Gh$$

In practice, bad first-stage decisions on storage budgets can lead to infeasibilities

We add slack variables, penalized in the objective, to remove these infeasibilities

Infeasibilities

$$\min_{\substack{p,ls,ph,e\geq 0\\ ne,f,\varphi\\ pe^{+/-}\geq 0}} \sum_{g\in G}\sum_{t\in T_k} MC_g\cdot p_{t,g,\omega} + VOLL\sum_{t\in T_k} ls_{t,\omega} + \boxed{PEN\sum_{g\in Gh}\left(pe^{+}_{g,t_1-1,\omega} + pe^{-}_{g,t_1-1,\omega} + pe^{+}_{g,t_2,\omega} + pe^{-}_{g,t_2,\omega}\right)}$$

$$p_{g,t,\omega} \leq x_g \quad \forall g\in G, t\in T_k$$

$$D_{t,\omega} - \textstyle\sum_{g\in G} p_{t,g,\omega} - ls_{t,\omega} + ne_{t,\omega} - \sum_{g\in Gh} ph_{t,g,\omega} = 0 \quad \forall t\in T_k$$

$$ne_{t,\omega} = \textstyle\sum_{l=(n,\cdot)} f_{l,t,\omega} - \sum_{l=(\cdot,n)} f_{l,t,\omega} \quad \forall t\in T_k$$

$$f_{l,t,\omega} = B_k\cdot\left(\theta_{n,t,\omega} - \theta_{n,t,\omega}\right)\ \forall l=(n,m)\in K, t\in T_k$$

$$F_l \leq f_{l,t,\omega} \leq F_l \quad \forall l=(n,m)\in K, t\in T_k$$

$$\sum_{g\in G}\sum_{t\in T_k} p_{t,g,\omega} = q_{k,\omega}$$

$$e_{g,t,\omega} = e_{g,t-1,\omega} - ph_{g,t,\omega} + A_{g,t,\omega} \quad \forall g\in Gh, t\in T_k$$

$$e_{g,t,\omega} \leq E_{g,t,\omega} \quad \forall g\in Gh, t\in T_k$$

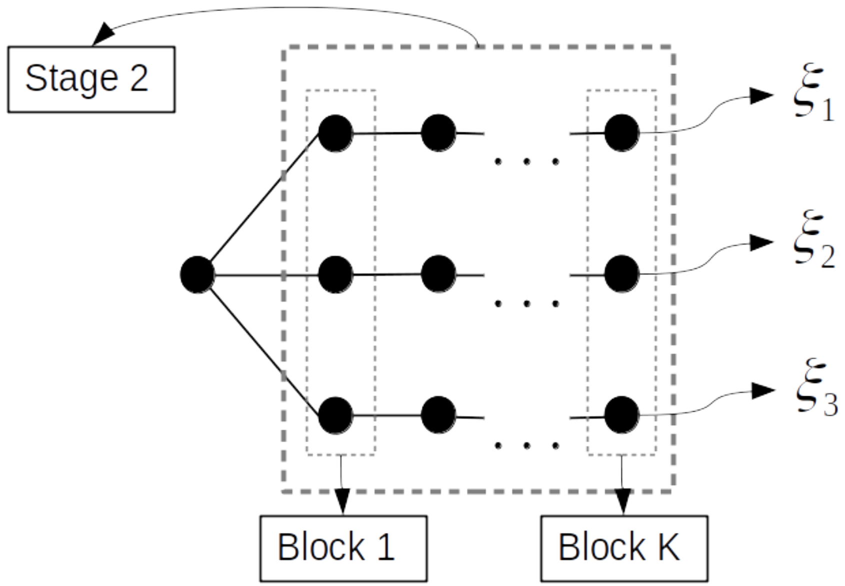
$$\boxed{\begin{aligned} e_{g,t_1-1,\omega} &= \hat{e}_{g,t_1-1,\omega} + pe^{+}_{g,t_1-1,\omega} - pe^{-}_{g,t_1-1,\omega}\ \forall g\in Gh\\ e_{g,t_2,\omega} &= \hat{e}_{g,t_2,\omega} + pe^{+}_{g,t_2,\omega} - pe^{-}_{g,t_2,\omega}\ \forall g\in Gh \end{aligned}}$$

One could consider adding feasibility cuts

- The construction of feasibility cuts requires the unboundedness ray
- With Gurobi this is available through the simplex algorithm
- Due to the scale of the problem, simplex is very slow
- We have opted for barrier

Subproblem structure

- The overall problem is decomposed into an array of subproblems
- The master problem decides on the investment and budget variables
- Several subproblems per scenario
  - Instead of an economic dispatch per scenario, we have the time horizon distributed across several subproblems



Master model mathematical description

$$\min_{x,q,e,\theta\geq 0} \sum_{g\in G} I_g \cdot x_g + E[\theta_\omega] + \rho_1 \cdot \left||x - x^v|\right|^2 + \rho_2 \cdot \left||q - q^v|\right|^2 + \rho_3 \cdot \left||e - e^v|\right|^2$$

$$x_g \leq X_g \quad \forall g \in G$$

$$E\left[\sum_k q_{k,\omega}\right] = TV$$

$$\boxed{(\theta_\omega, x_g, q_{k,\omega}, e_{g,ti,\omega}) \in Opt(\omega)}$$

$$e_{g,ti,\omega} \leq E_{g,ti,\omega} \ \ \forall g \in Gh, ti \in T_k$$

The storage at the boundaries becomes a first-stage decision

Master model mathematical description

$$\min_{x,q,e,\theta\geq 0} \sum_{g\in G} I_g \cdot x_g + E[\theta_\omega] + \rho_1 \cdot \left||x - x^v|\right|^2 + \rho_2 \cdot \left||q - q^v|\right|^2 + \rho_3 \cdot \left||e - e^v|\right|^2$$

$$x_g \leq X_g \quad \forall g \in G$$

$$E\left[\sum_k q_{k,\omega}\right] = TV$$

$$(\theta_\omega, x_g, q_{k,\omega}, e_{g,ti,\omega}) \in \boxed{Opt(\omega)}$$

$$e_{g,ti,\omega} \leq E_{g,ti,\omega} \quad \forall g \in Gh, ti \in T_k$$

The storage at the boundaries becomes a first-stage decision

The storage at the boundaries is constrained by optimality cuts

Master model mathematical description

$$\min_{x,q,e,\theta\geq 0} \sum_{g\in G} I_g \cdot x_g + E[\theta_\omega] + \rho_1 \cdot \left|\left|x - x^v\right|\right|^2 + \rho_2 \cdot \left|\left|q - q^v\right|\right|^2 + \rho_3 \cdot \left|\left|e - e^v\right|\right|^2$$

$$x_g \leq X_g \quad \forall g \in G$$

There are budget variables for each scenario and temporal subdivision

$$E\left[\sum_k q_{k,\omega}\right] = TV$$

The storage at the boundaries becomes a first-stage decision

$$(\theta_\omega, x_g, q_{k,\omega}, e_{g,ti,\omega}) \in \boxed{Opt(\omega)}$$

$$e_{g,ti,\omega} \leq E_{g,ti,\omega} \quad \forall g \in Gh, \mathrm{t_i} \in T_k$$

The storage at the boundaries is constrained by optimality cuts

Master model mathematical description

$$\min_{x,q,e\geq 0,\theta} \sum_{g\in G} I_g \cdot x_g + E[\theta_\omega] + \rho_1 \cdot ||x - x^v||^2 + \rho_2 \cdot ||q - q^v||^2 + \rho_3 \cdot ||e - e^v||^2$$

The storage is regularized as well

$$x_g \leq X_g \quad \forall g \in G$$

There are budget variables for each scenario and temporal subdivision

$$E\left[\sum_k q_{k,\omega}\right] = TV$$

The storage at the boundaries becomes a first-stage decision

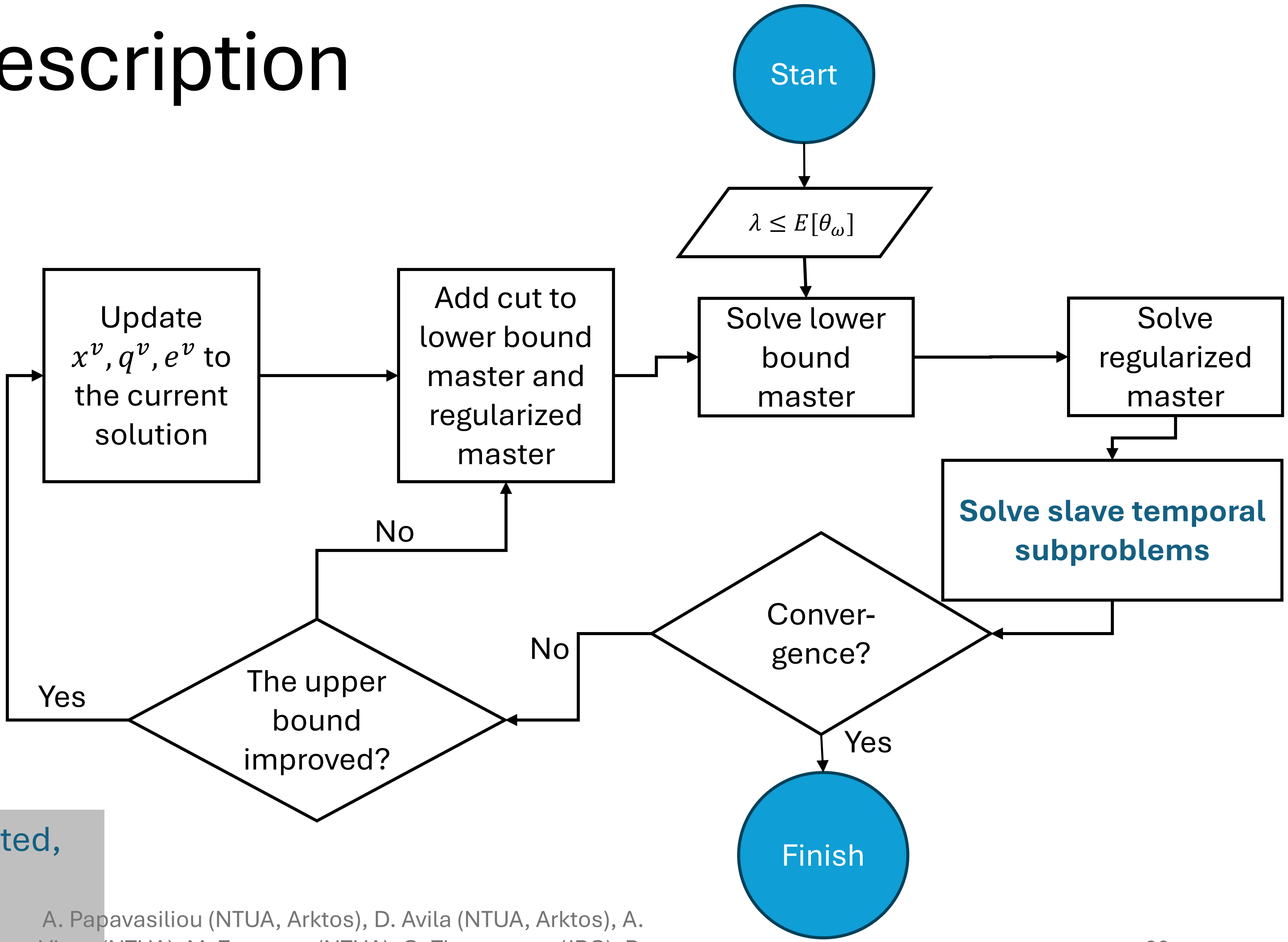
$$(\theta_\omega, x_g, q_{k,\omega}, e_{g,ti,\omega}) \in \boxed{Opt(\omega)}$$

$$e_{g,ti,\omega} \leq E_{g,ti,\omega} \quad \forall g \in Gh, \mathrm{t_i} \in T_k$$

The storage at the boundaries is constrained by optimality cuts

Algorithmic description

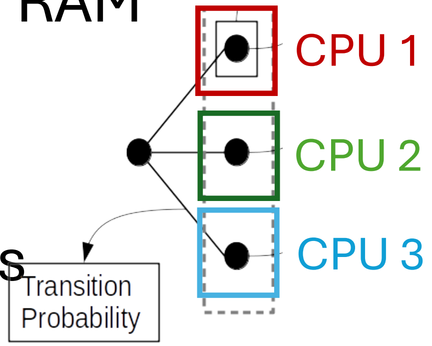
1. Provide initial lower bound value for $E[\theta_\omega]$, and initial x^v, q^v, e^v estimates
2. Solve lower bound master problem
3. Solve regularized master problem
4. **Solve slave temporal subproblems**
5. Update masters' problems



The main steps remain unaffected, the difference comes on the subproblems

Limitations

- In practice the convergence can be slow
  - Initial iterations are not particularly intelligent, because value functions are bad approximations of second-stage costs
  - The iterate tends to waste time exploring the corners of the feasible space
- Subproblems are large-scale LPs
  - Iterations are costly
  - A full EU system at nodal level becomes restrictive due to RAM considerations
- Limitations of high-performance computing
  - The number of CPUs is limited by the number of scenarios



Algorithmics

Decoupling target volume constraint

Scenario decomposition

Regularization

Temporal decomposition

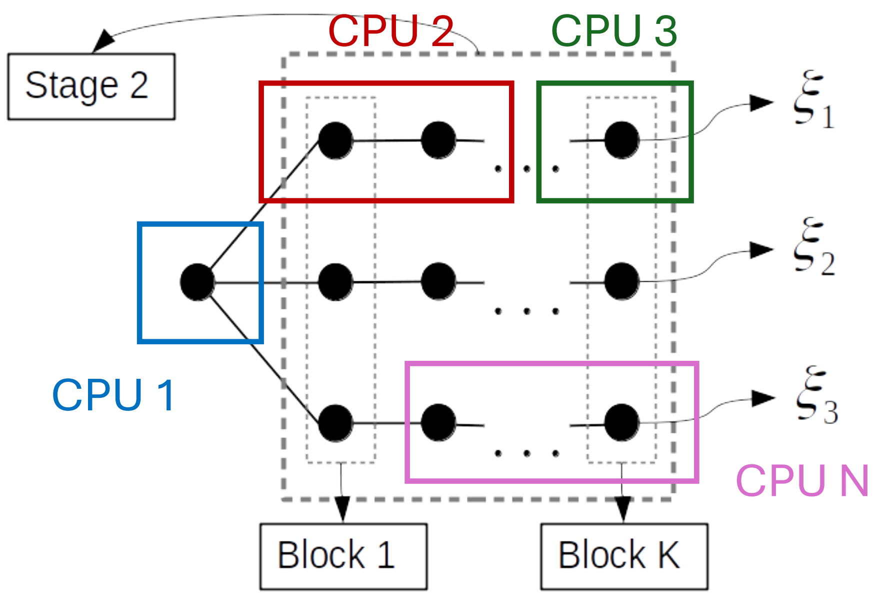
Parallel computing implementation

High performance computing

- The temporal subproblem structure allows us to enhance the applicability of high-performance computing
- The number of subproblems increases from $|\Omega|$ to $|\Omega| \cdot K$
  - Here K is the number of temporal blocks
- For example, within the considered system $|\Omega| = 15$
  - This limits the number of CPUs to 15, when using traditional decomposition schemes
  - If we let $K = 52$, so that each subproblem is a week, the proposed approach can use up to 780 CPUs

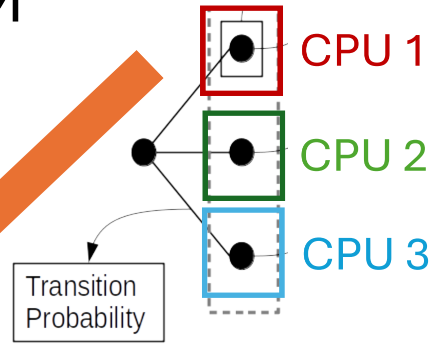
Distributed computing algorithm

- The largest part of the computational work comes down to optimizing the subproblems
- The master problem is handled by the master process
- The slave subproblems are distributed among the available CPUs
  - Each CPU oversees an array of subproblems
  - During each iteration, each CPU optimizes this array of subproblems



Limitations

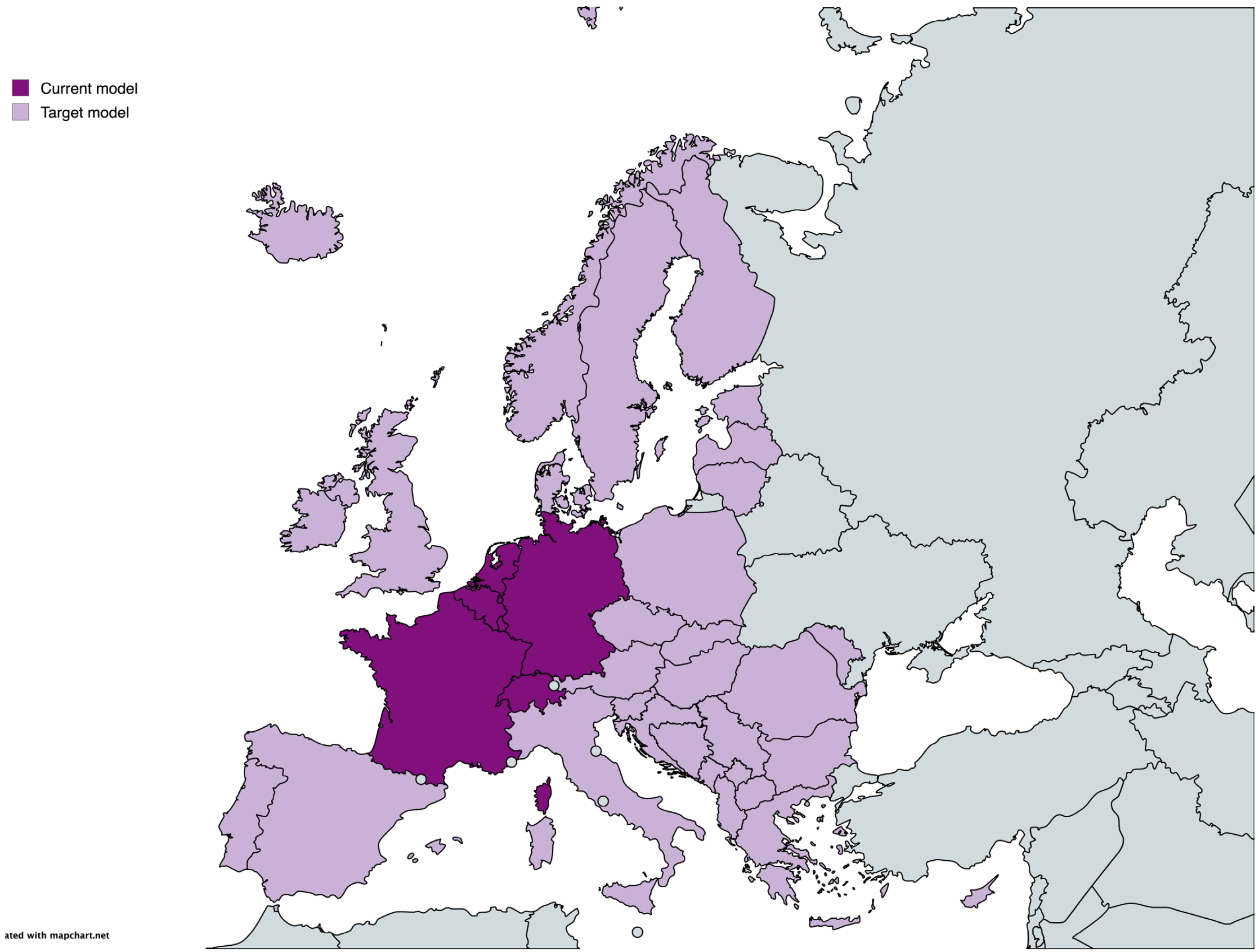
- In practice the convergence can be slow
  - Initial iterations are not particularly intelligent, because value functions are bad approximations of second-stage costs
  - The iterate tends to waste time exploring the corners of the feasible space
- Subproblems are large-scale LPs
  - Iterations are costly
  - A full EU system at nodal level becomes restrictive due to RAM considerations
- Limitations of high-performance computing
  - The number of CPUs is limited by the number of scenarios



Results

Test system

- We consider the CWE system
  - France, Germany, Luxembourg, Belgium, the Netherlands, Switzerland
  - Represents 40% of the full EU load



System characteristics

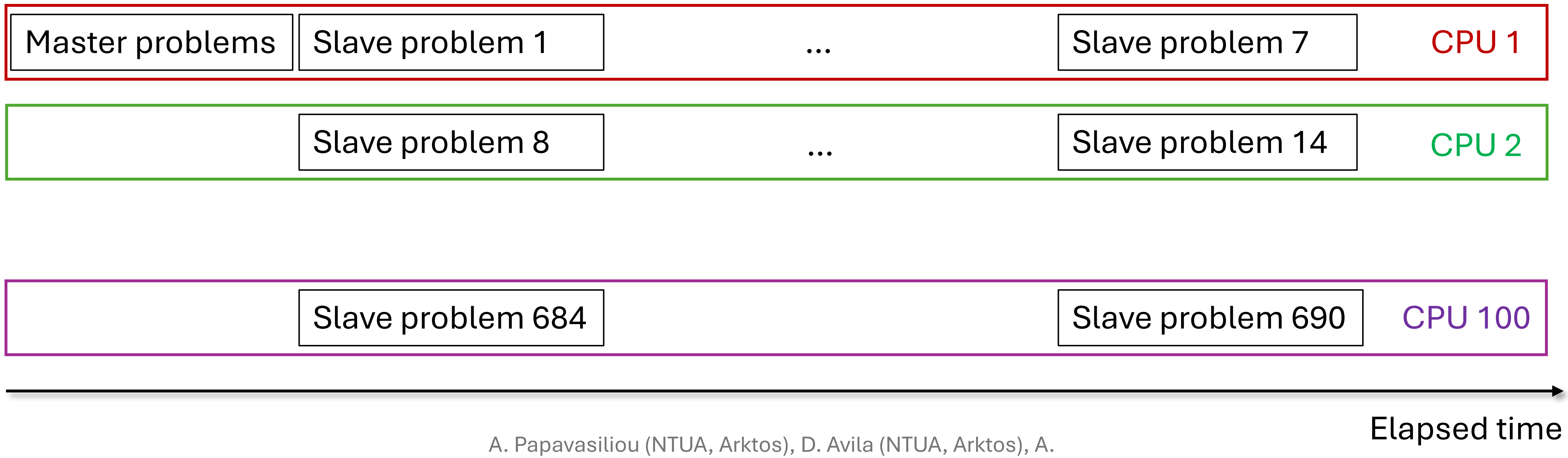
	Characteristics	Data source
Transmission network	• Nodal model • 1549 nodes • B-theta linearization of the power flow equations	• PyPSA
Thermal generation	• Unit-based model • 918 generation units • No unit commitment of generators	• PyPSA
Hydrological generation	• Unit-based model • 3 technologies: run-of-river, reservoir, pumped storage • 121 reservoirs, 66 pumped storage units, 367 run-of-river units	• PyPSA
Solar and wind generation	• 1456 solar and 1417 wind generation units	• PyPSA • ENTSO-E transparency platform
Uncertainty	• 15 climate scenarios	• European resource adequacy assessment (ERAA)
Time horizon	• One target year	

Hardware specifications

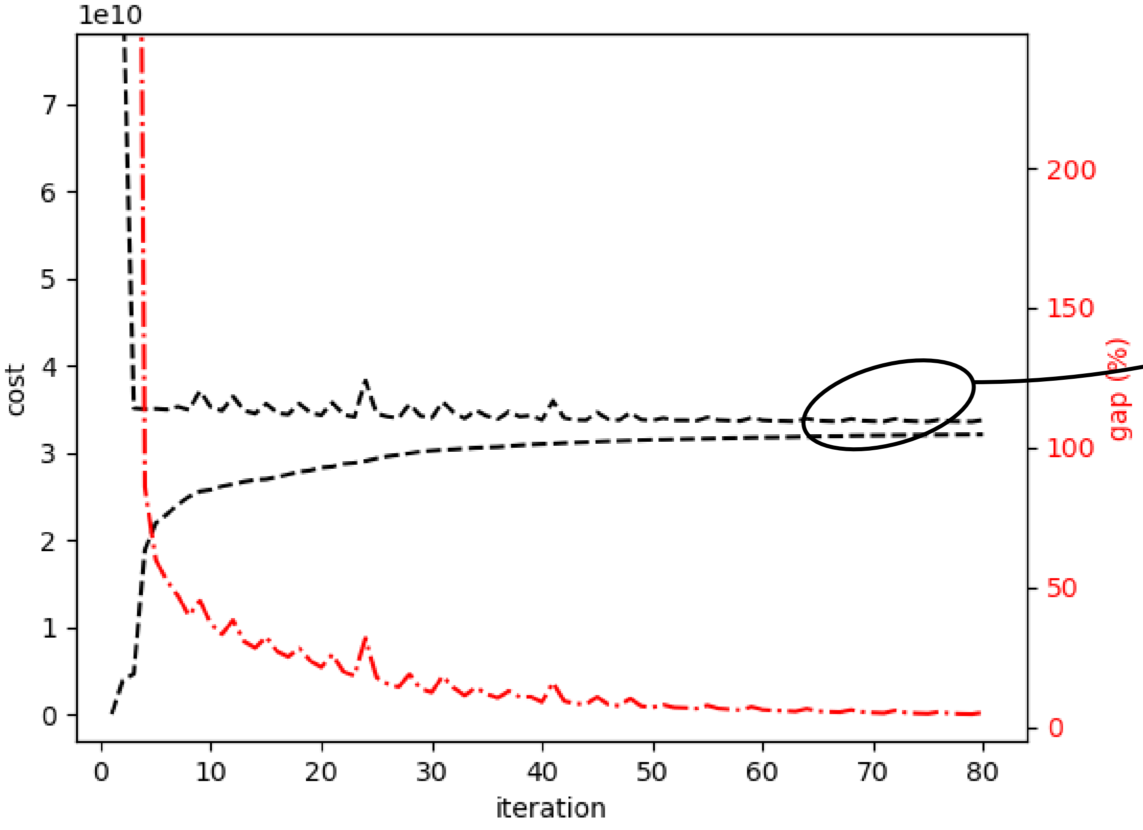
	Characteristic
Number of CPUs	100
Number of nodes	25
RAM per node	56 GB
Slave subproblems per CPU	+/- 7

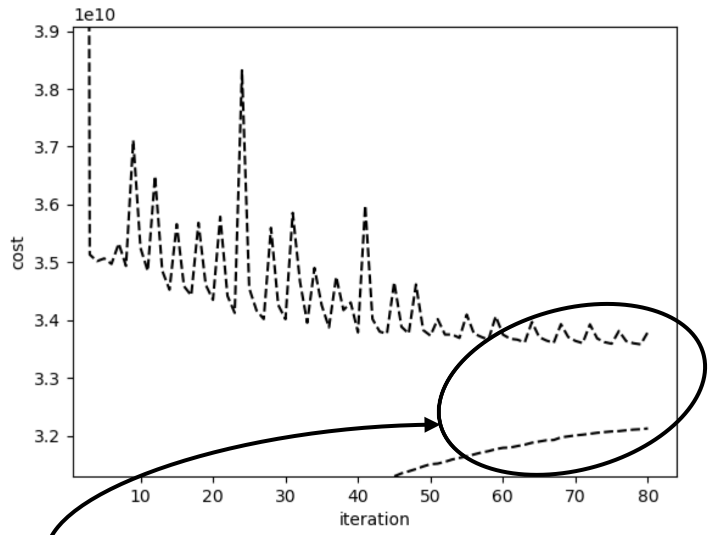
Typical iteration structure

- There are 15 scenarios
- The economic dispatch is partitioned into 46 subproblems
  - This gives a total of 15*46= 690 slave subproblems
  - As we have 100 CPUs, roughly 7 subproblems per CPU



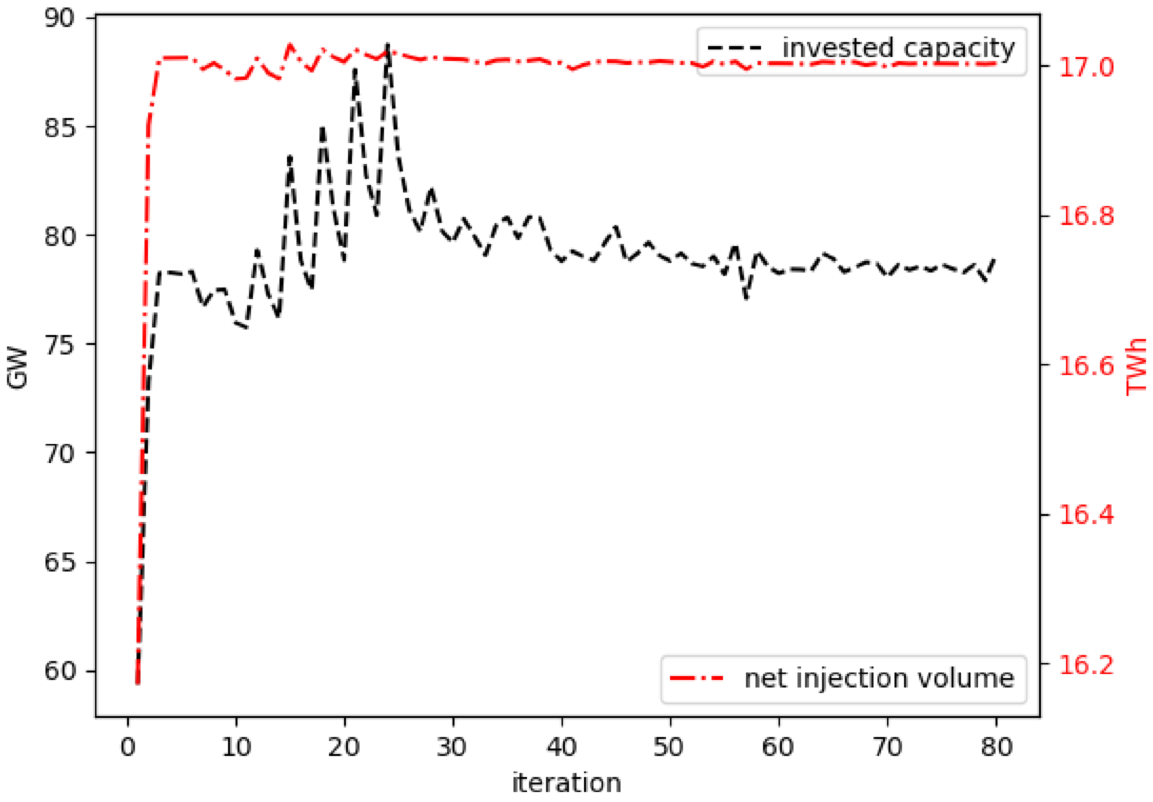
Convergence evolution





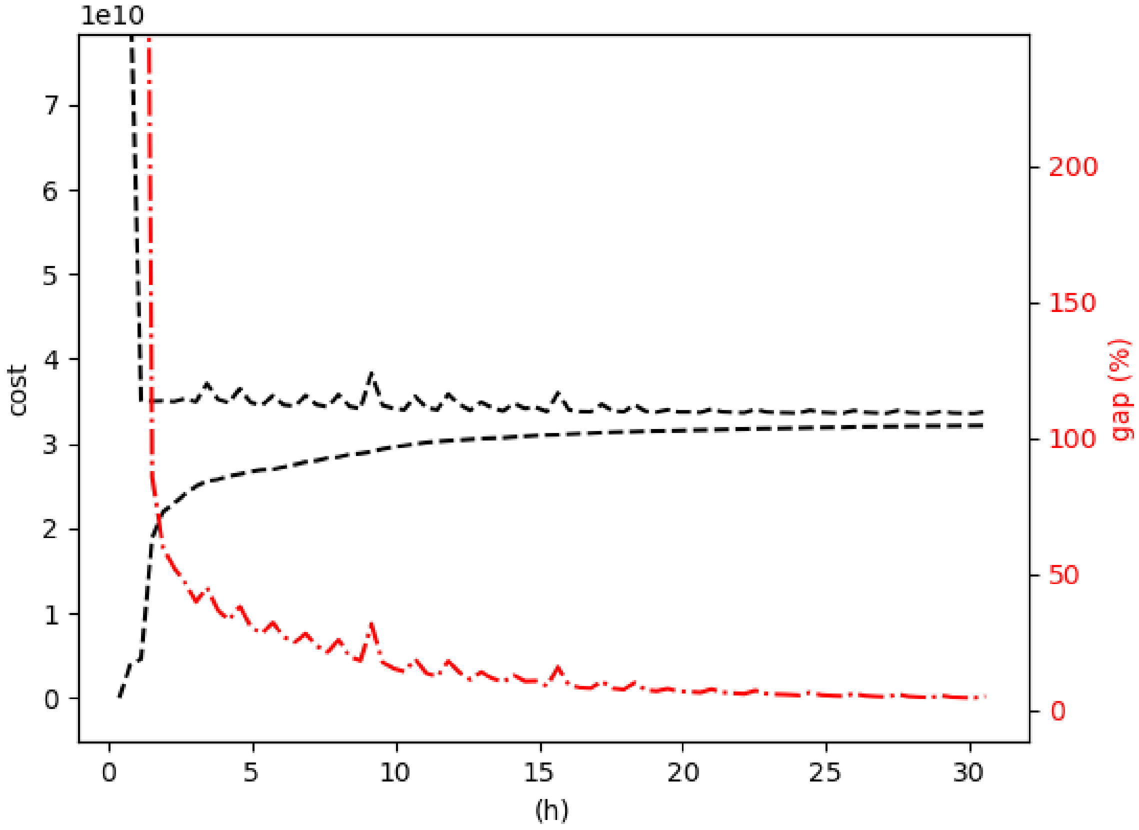
- The algorithm runs over 80 iterations
  - 3% optimality gap
- The upper bound stabilizes after 60 iterations
  - We observe just lower bound improvements

Evolution of investments



- During early iterations, the invested capacity presents an unstable behavior
  - At this stage, the value function approximation is poor
  - The algorithm spends time building a value function approximation so as to provide reasonable volume and storage “budgets” for the subproblems
- After some tens of iterations, we observe a more stable behavior
  - At this stage, the value function steers reasonable “budgets” to feed the subproblems
  - The algorithm proceeds to refine the investment decisions
- The figure is complemented with the net injected volume, which seems to converge to the target

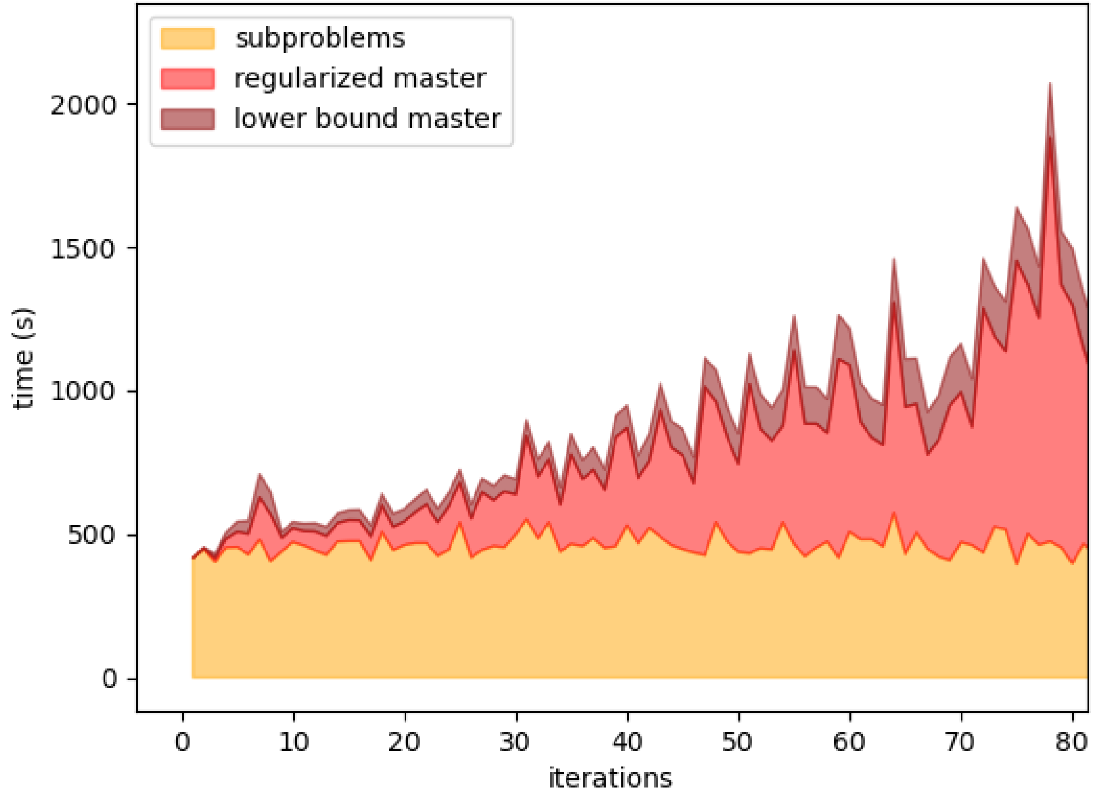
Elapsed time



The algorithm runs over 30 hours

- The average time per iteration is 20 minutes

Key run time statistics



- During early iterations, the subproblems take up most of the computing time
- Towards the end of execution, the regularized master problem starts taking the most time
  - Cut selection?

Expected value of perfect information (EVPI)

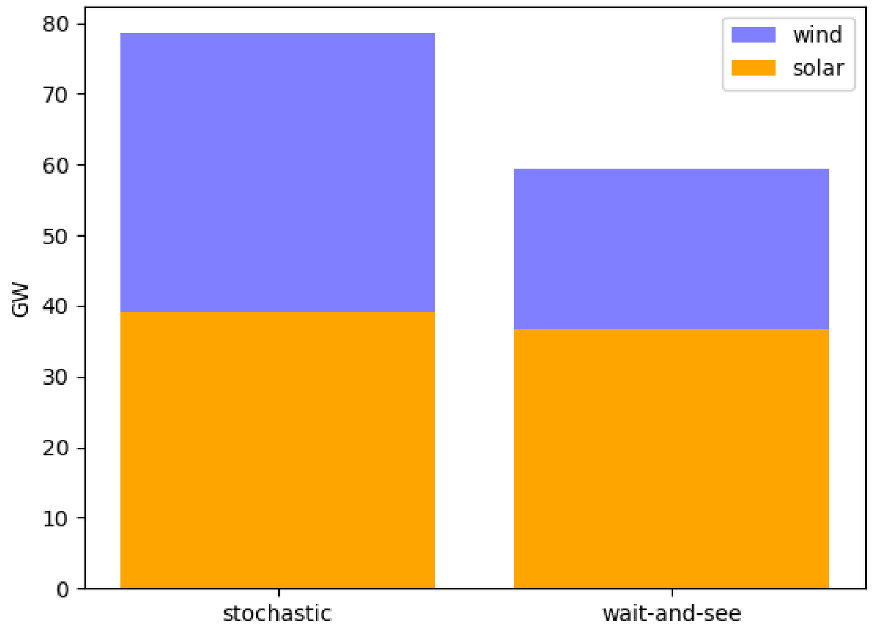
- We compute $EVPI$, i.e. the difference between the total cost of the wait-and-see solution (WS) and the stochastic solution (*SP*)
  - Theory guarantees that:

$$WS = E_{\omega \in \Omega}\left[\min_{x} z(x,\omega)\right] \leq \min_{x} E_{\omega \in \Omega}[z(x,\omega)] = SP$$

  - A low $EVPI$ value suggests limited added value of non-anticipativity
- The stochastic solution is 3.6% higher than the wait-and-see solution
  - The system costs would reduce by 3.6% if we could perfectly anticipate the future

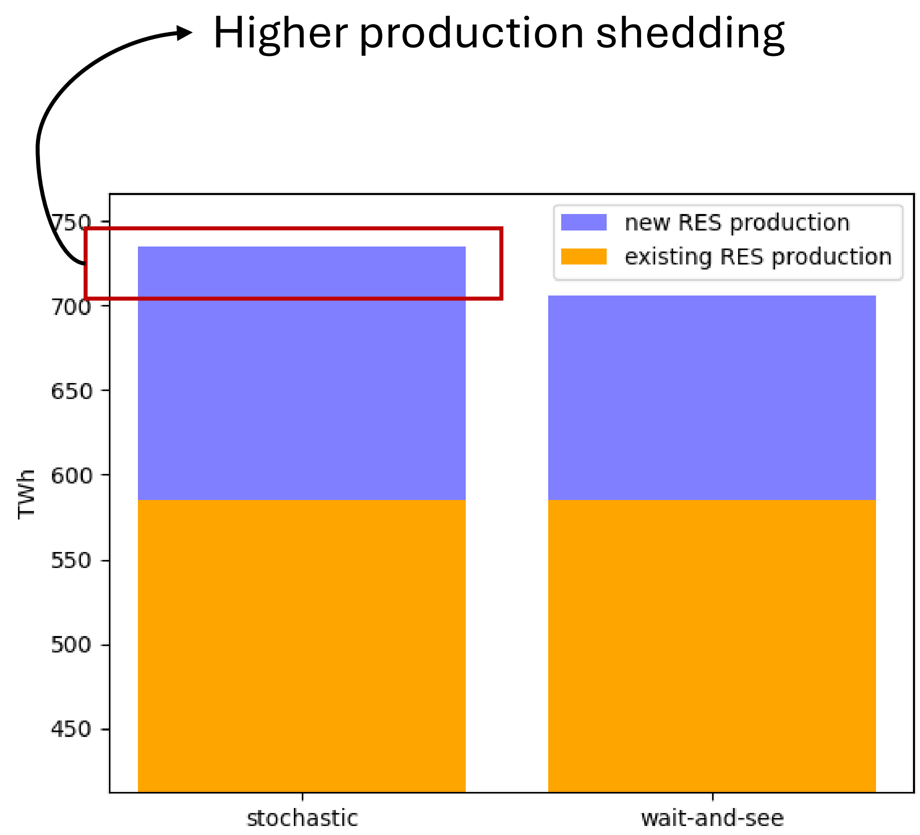
Solution differences

- The figure presents the differences in new installed capacity
- The stochastic solution is more conservative
  - It builds more wind to protect itself from uncertain events
  - Solar capacity appears less affected by the effect of incomplete information
- The wait-and-see solution strategy is less aggressive in investment
  - It is able to perfectly estimate the conditions



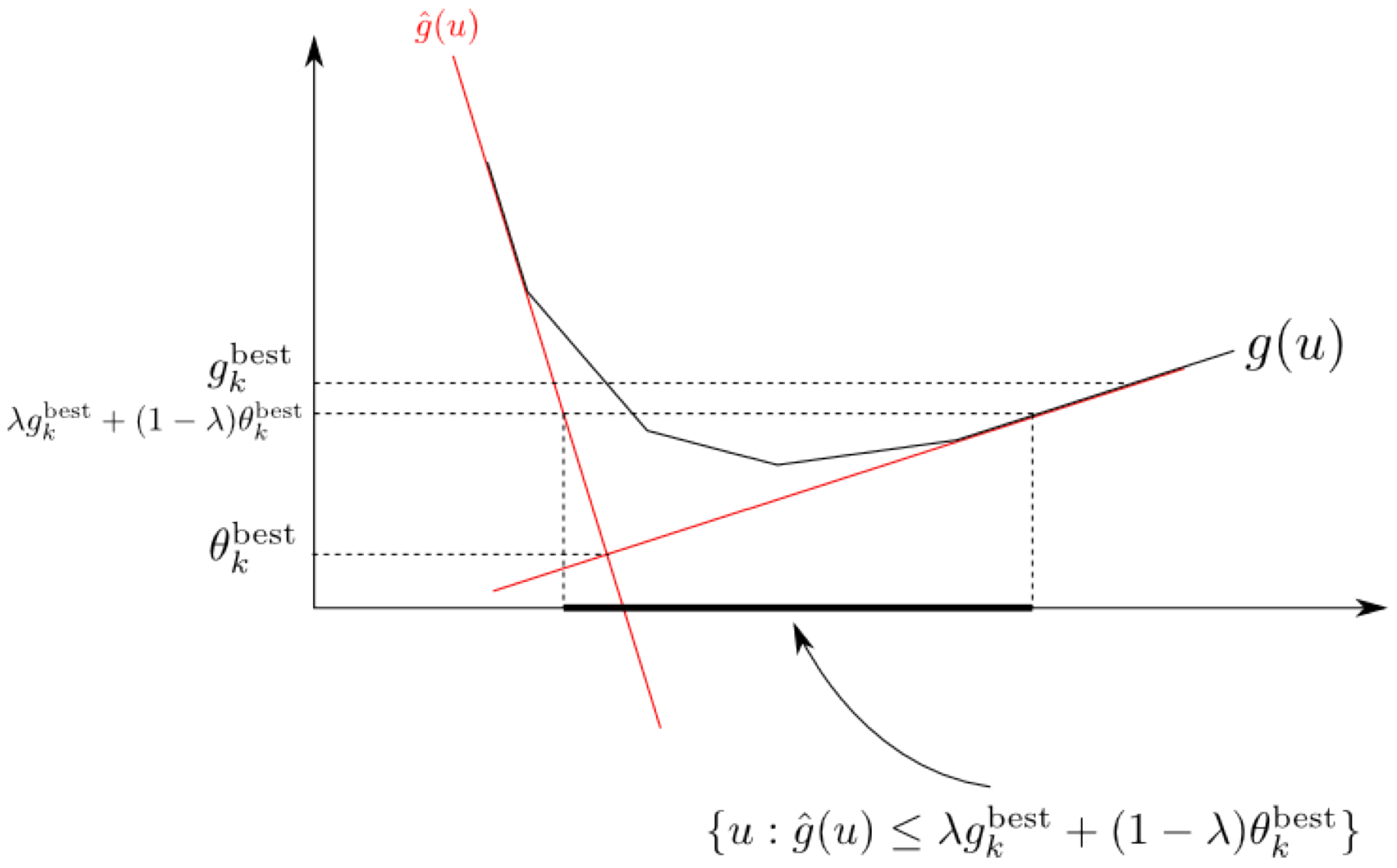
Solution differences (II)

- The figure presents the difference in production volumes
- We observe differences in the production of new RES volumes
  - However, both solutions face the same target volume
  - The excess volume is due to production shedding
  - The stochastic solution is prone to higher RES curtailments



Ongoing Extensions

Visual description of level set algorithm



Level-set stabilization

NOMENCLATURE

- I: set of candidate investment projects
- S: set of scenarios
- B: set of temporal blocks
- H: set of hydro units

Level-set stabilization

NOMENCLATURE

- x_i: Investment decisions of project $\mathrm{i} \in I$
- $\mathrm{q}_{b,s}$: Budget variables of RES injection for $\mathrm{b} \in \mathrm{B}$, s∈S
- $\mathrm{e}_{h,t_b,s}$: End-of-block storage volume for hydro $\mathrm{h} \in \mathrm{H}$, $\mathrm{b} \in \mathrm{B}$, s∈S
- $\theta_{b,s}$: Epigraph variable approximating second-stage cost for b∈B and scenario s∈S

Level-set stabilization

NOMENCLATURE

- T_b^-: Time before each block b∈B
- T_b^+: End time of block b∈B
- c_i: Investment cost of project i $\in I$
- E_h^{max}: Storage capacity limit for hydro h ∈ H
- TV: Target value of RES injection
- $|S|$: number of scenarios
- $\Phi_{b,s}(x, q, e)$: operational cost for block b∈B and scenario s∈S
- λ: Dual multipliers from the second-stage problems

Master Problem mathematical formulation

MASTER PROBLEM

(MP):

$$LB^{(j)} = minimise_{\theta_{b,s}, x_i, q_{b,s}, e_{h,tb,s} \geq 0} \sum_{i \in I} c_I^T x_i + \frac{1}{|S|} \sum_{s \in S} \sum_{b \in B} \theta_{b,s}$$

s.t.:

$$\theta_{b,s} \geq \Phi_{b,s}^{(j)}(x, q, e) + \sum_{i \in I} {\lambda_{i,b,s}^{x}}^{(j)} \left(x_i - x_i^{(j)}\right) + {\lambda_{b,s}^{q}}^{(j)} \left(q_{b,s} - q_{b,s}^{(j)}\right)$$

$$+ 1_{\{T_b^- < 1\}} \sum_{h \in H} {\lambda_{h,b,s}^{e_{T_b^-}}}^{(j)} \left(e_{h,T_b^-,s} - e_{h,T_b^-,s}^{(j)}\right)$$

$$+ 1_{\{T_b^+ < T\}} \sum_{h \in H} {\lambda_{h,b,s}^{e_{T_b^+}}}^{(j)} \left(e_{h,T_b^+,s} - e_{h,T_b^+,s}^{(j)}\right), \forall j$$

$$\in \{1, \ldots p, \ldots, |J|\}, b \in B, s \in S$$

$$\frac{1}{|S|} \sum_{s \in S} \sum_{b \in B} q_{b,s} = TV$$

$$e_{h,tb,s} \leq E_h^{max}, \forall h \in H, b \in B, s \in S$$

Level-set problem mathematical formulation

LEVEL-SET PROBLEM

(LSP):

$$minimise_{\theta_{b,s},x_i,q_{b,s},e_{h,tb,s}\geq 0}\left\{\sum_{i\in I}\|x_i - x_i^*\|_2^2\right.$$
$$+\sum_{s\in S}\sum_{b\in B}\left[\left\|q_{b,s}-q_{b,s}^*\right\|_2^2\right.$$
$$\left.\left.+\left(\sum_{h\in H}\left\|e_{h,T_b^-,s}-e_{h,T_b^-,s}^*\right\|_2^2+\left\|e_{h,T_b^+,s}-e_{h,T_b^+,s}^*\right\|_2^2\right)\right]\right\}$$

s.t.:

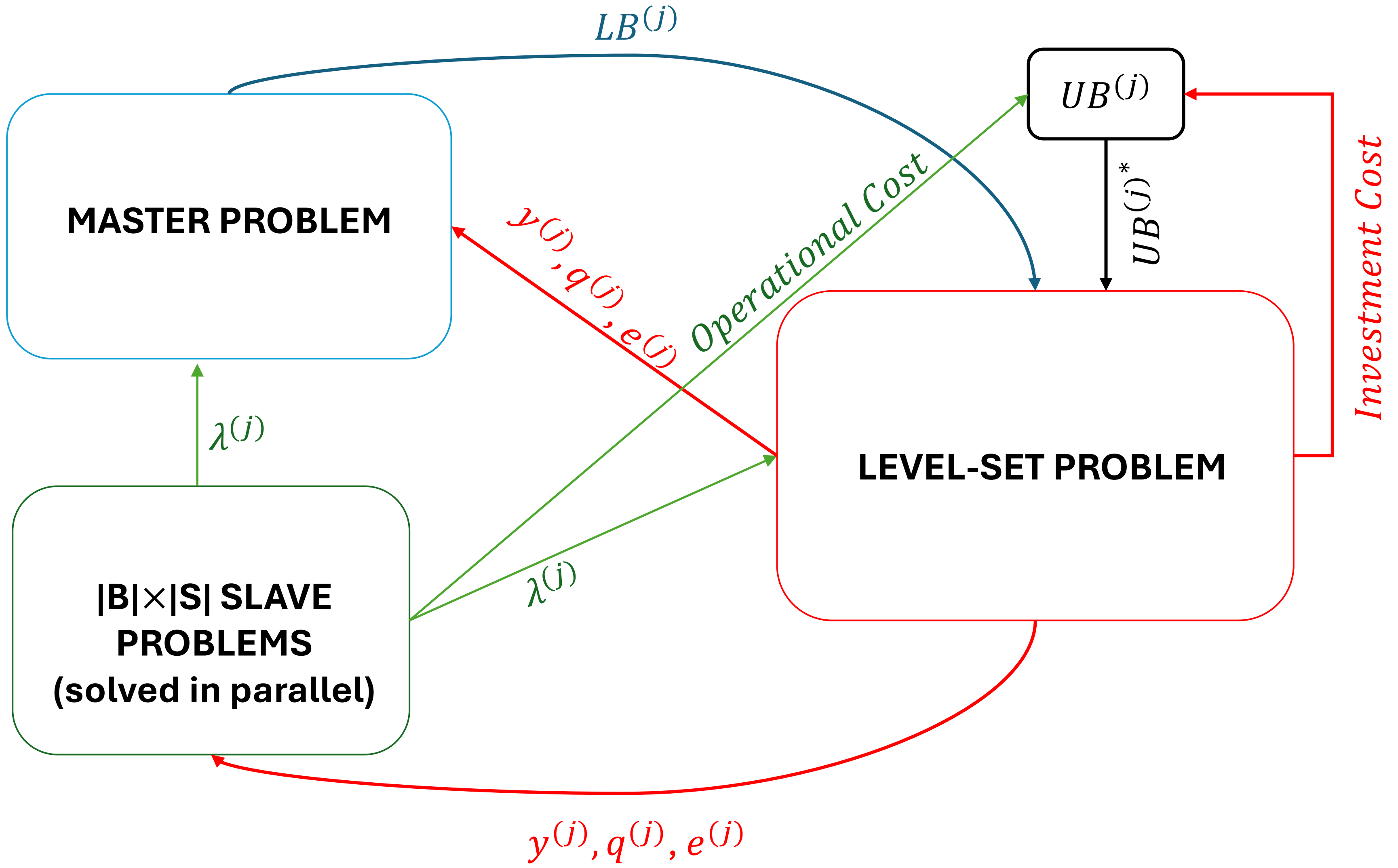
$$\theta_{b,s} \geq \Phi_{b,s}^{(j)}(x,q,e)+\sum_{i\in I}\lambda_{i,b,s}^{x}{}^{(j)}\left(x_i - x_i^{(j)}\right)+\lambda_{b,s}^{q}{}^{(j)}\left(q_{b,s}-q_{b,s}^{(j)}\right)$$
$$+1_{\{T_b^-<1\}}\sum_{h\in H}\lambda_{h,b,s}^{e_{T_b^-}}{}^{(j)}\left(e_{h,T_b^-,s}-e_{h,T_b^-,s}^{(j)}\right)$$
$$+1_{\{T_b^+<T\}}\sum_{h\in H}\lambda_{h,b,s}^{e_{T_b^+}}{}^{(j)}\left(e_{h,T_b^+,s}-e_{h,T_b^+,s}^{(j)}\right),\forall j$$
$$\in\{1,\ldots p,\ldots,|J|\}, b\in B, s\in S$$

$$\frac{1}{|S|}\sum_{s\in S}\sum_{b\in B}q_{b,s}=TV$$

$$e_{h,tb,s}\leq E_h^{max}, \forall h\in H, b\in B, s\in S$$

$$\sum_{i\in I}c_I^T x_i+\frac{1}{|S|}\sum_{s\in S}\sum_{b\in B}\theta_{b,s}\leq aLB^{(j)}+(1-a)UB^{(j)^*}$$

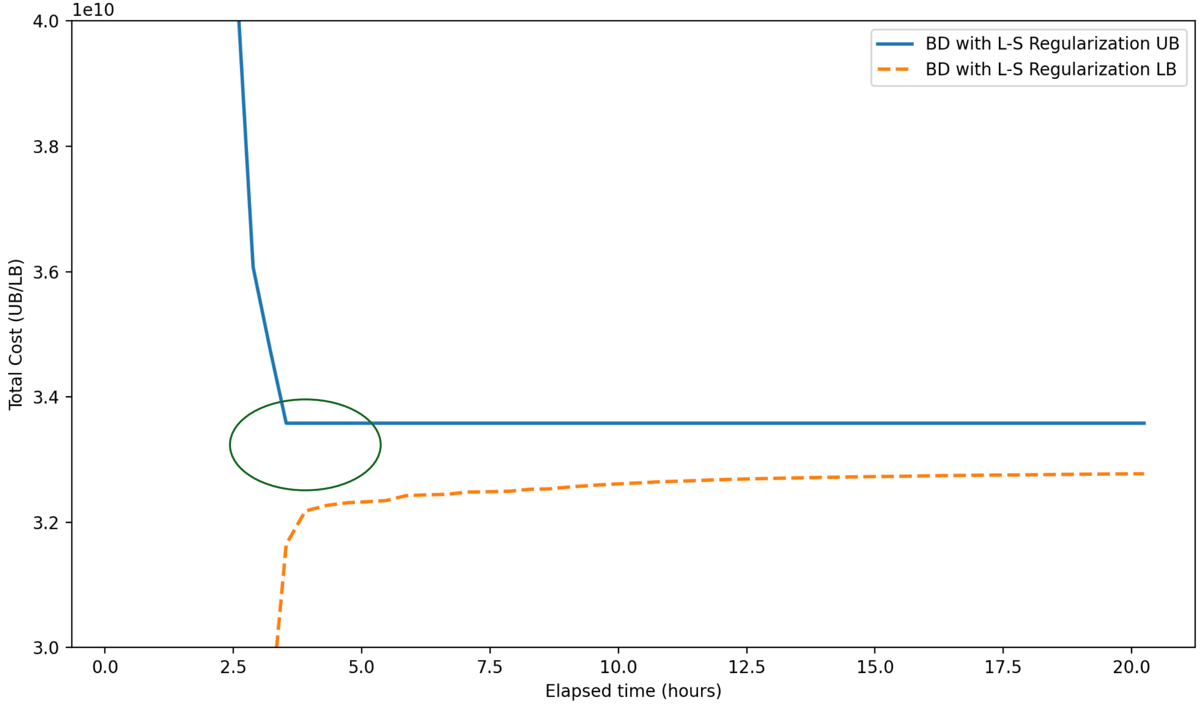
Level-set stabilization algorithm flowchart

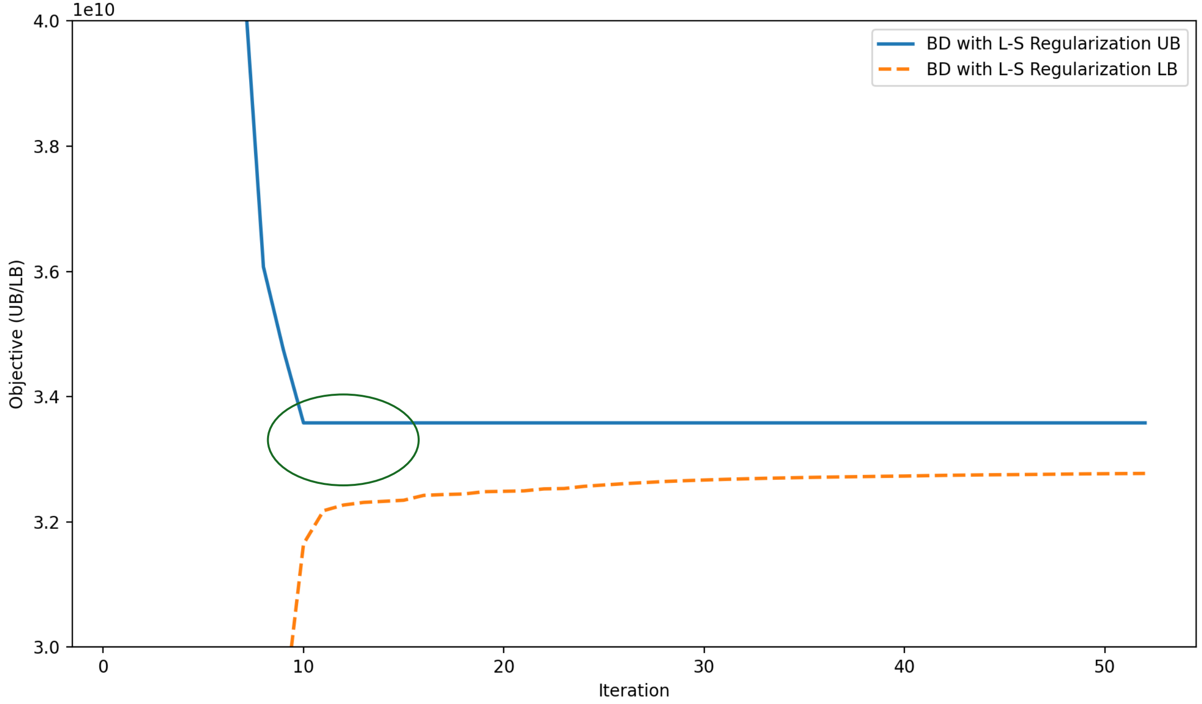


$$UB^{(j)^*} = min_{0 \le j \le |J|} UB^{(j)}$$

CWE results of the level-set stabilization algorithm (a = 0.5 & 14 subperiods per scenario)

Each scenario was partitioned into 14 subproblems to prevent master-problem blow-up



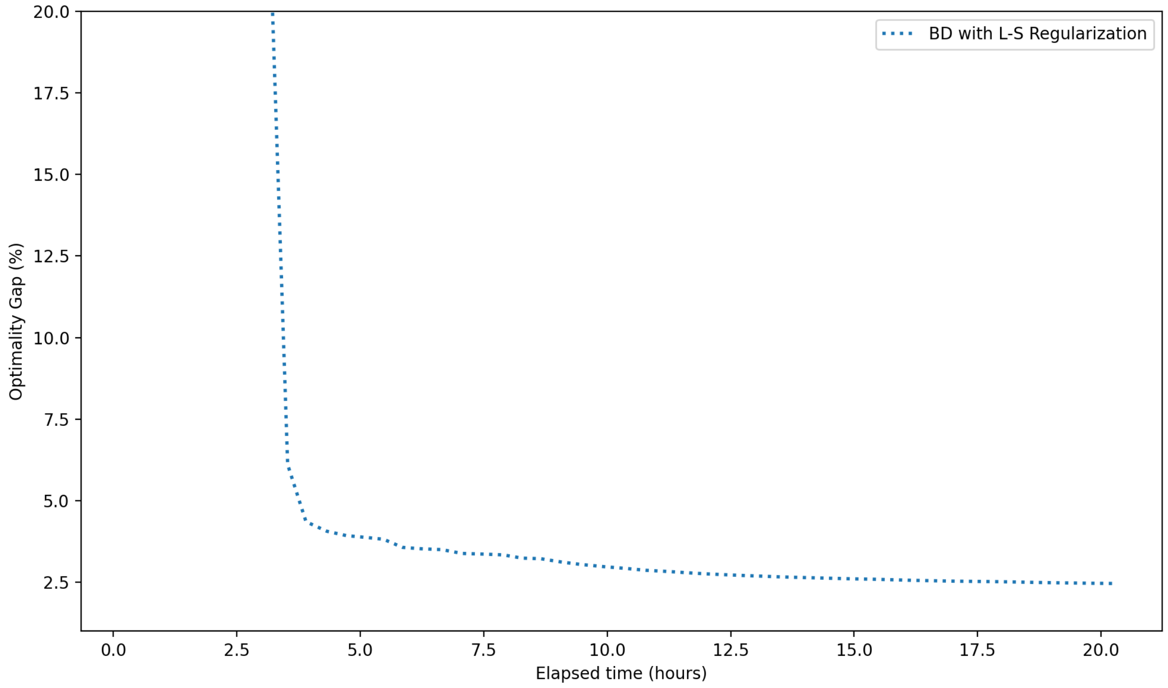


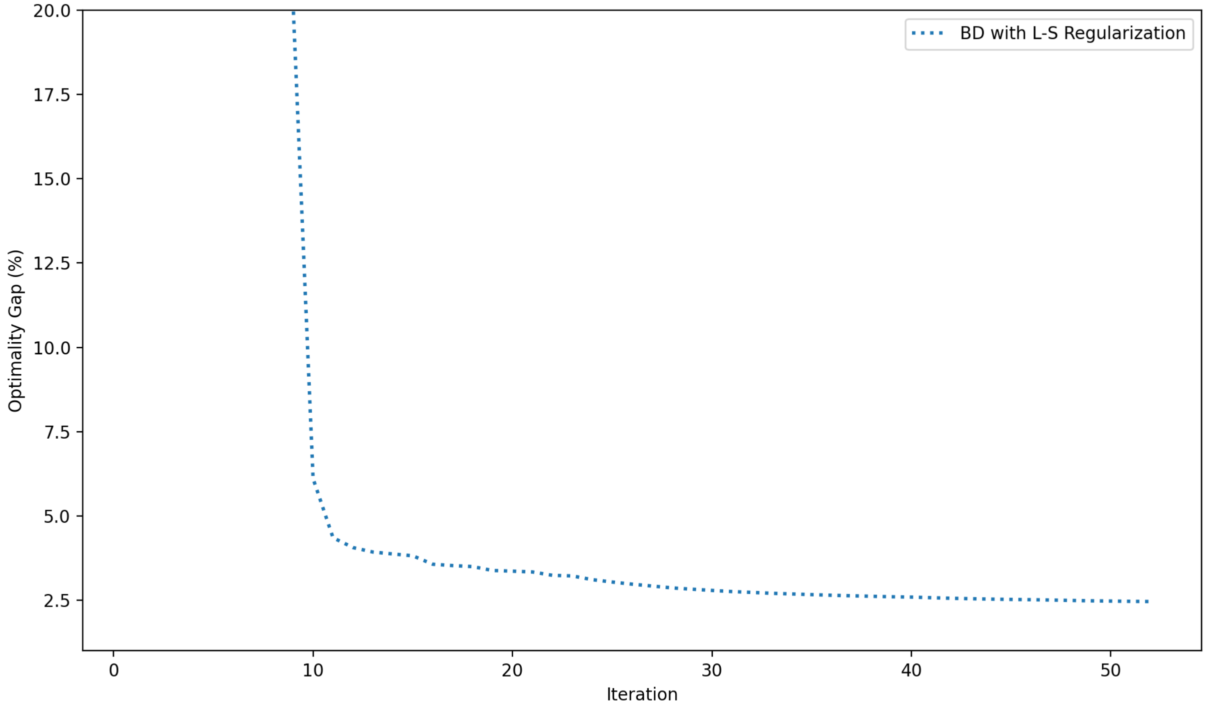
The upper bound stabilizes after 9 iterations (~3.5 hours)

We observe just lower bound improvements

Results of the level-set stabilization algorithm (a = 0.5 & 14 subperiods per scenario)

Optimality gap is ~2.46% after 20 hours and 50 iterations





Dynamic level-set parametrization [9]

- When the upper bound stalls for several iterations, it’s a sign the algorithm is circling around the same first-stage choices
- To address this, we adjust the level-set parameter “a” so the level set moves closer to the lower bound
- This tightens the feasible region, drives the UB down, and triggers more aggressive cut generation

Dynamic level-set parametrization algorithm

Step 0: Initialize parameters: $\alpha^{(1)}$, ω, r^{min}, r^{max}, a_{min}, a_{max}, a^{-}, a^{+}

Step 1: Compute level-set value:

$$L^{(j)} \longleftarrow \alpha^{(j-1)}LB^{(j)} + (1-\alpha^{(j-1)})UB^{(j)^*}$$

Step 2a: Compute $\alpha^{(j)}$ If Upper Bound remains the same for 2 subsequent iterations:

$$\alpha^{(j)} \longleftarrow max\{a_{min}, \min\left\{a^{-}, \frac{\alpha^{(j-1)}}{2}\right\}\}$$

Step 2b: Compute $\alpha^{(j)}$ If Lower Bound remains the same for 2 subsequent iterations:

$$\alpha^{(j)} \longleftarrow \min\{a_{max}, max\left\{a^{+}, 1-\frac{1-\alpha^{(j-1)}}{2}\right\}\}$$

Step 3: Compute Upper/ Lower Bound progress ratios:

$$r^{l} = \frac{LB^{(j)}-LB^{(j-1)}}{L^{(j)}-LB^{(j-1)}}, r^{u} = \frac{UB^{(j-1)}-UB^{(j)}}{UB^{(j-1)}-L^{(j)}}$$

Step 4: Compute $\alpha^{(j)}$ for the following cases:

If $(r^{u} \geq r^{max})\&\left(r^{l} \leq r^{min}\right)$, then: $\alpha^{(j)} \longleftarrow 1-\left(1-\alpha^{(j-1)}\right)\omega$

Else If $\left(r^{l} < r^{min}\right)\&\left(r^{u} < r^{min}\right)$, then: $\alpha^{(j)} \longleftarrow \alpha^{(j-1)}\omega^{2}$

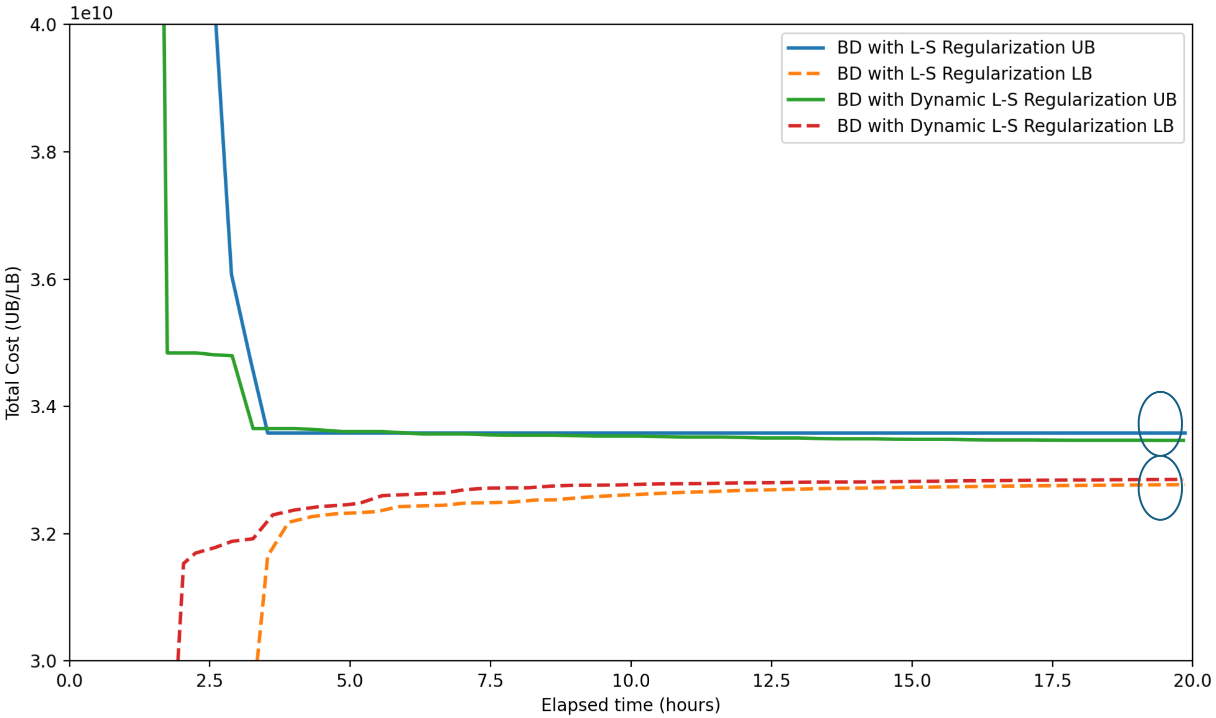
Else If $\left(r^{min} < r^{u} < r^{max}\right)\&\left(r^{l} < r^{min}\right)$, then:$\alpha^{(j)} \longleftarrow 1-\left(1-\alpha^{(j-1)}\right)\omega^{2}$

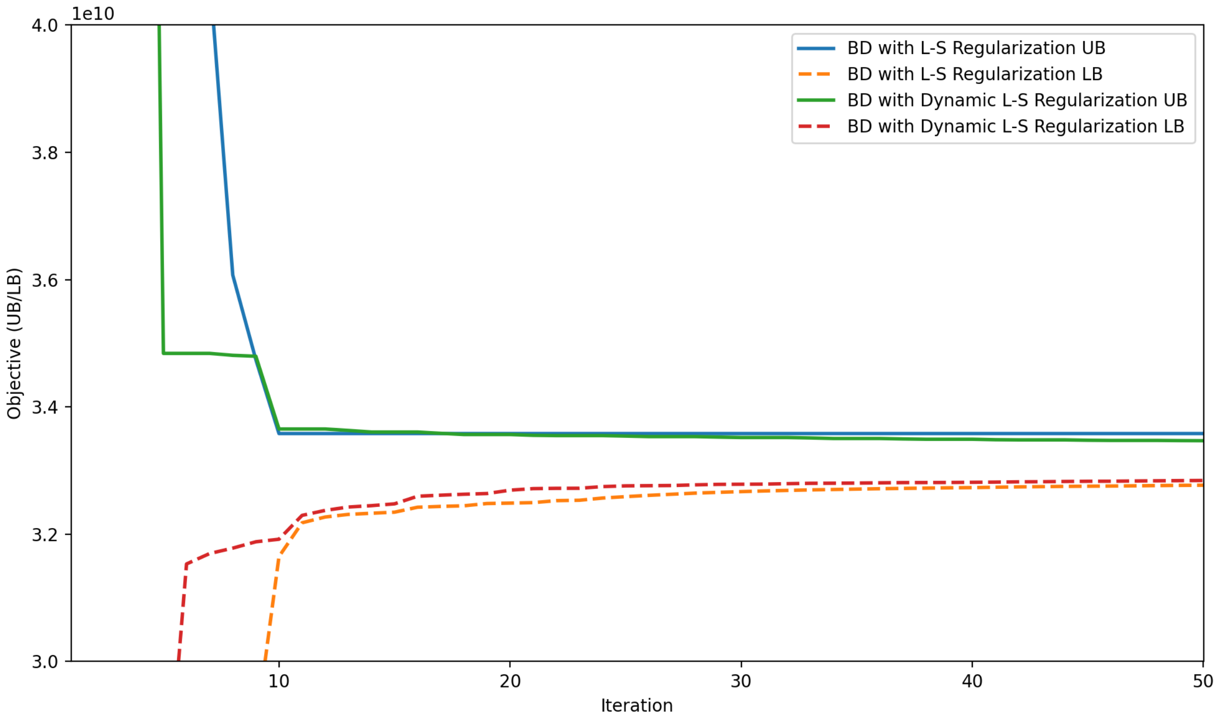
Else If $\left(r^{min} < r^{l} < r^{max}\right)\&\left(r^{u} < r^{min}\right)$, then:$\alpha^{(j)} \longleftarrow \alpha^{(j-1)}\omega$

Else: Do not change a

Step 5: Return $\alpha^{(j)} \longleftarrow \min\{a_{max}, \max\{a_{min}, \alpha^{(j)}\}\}$

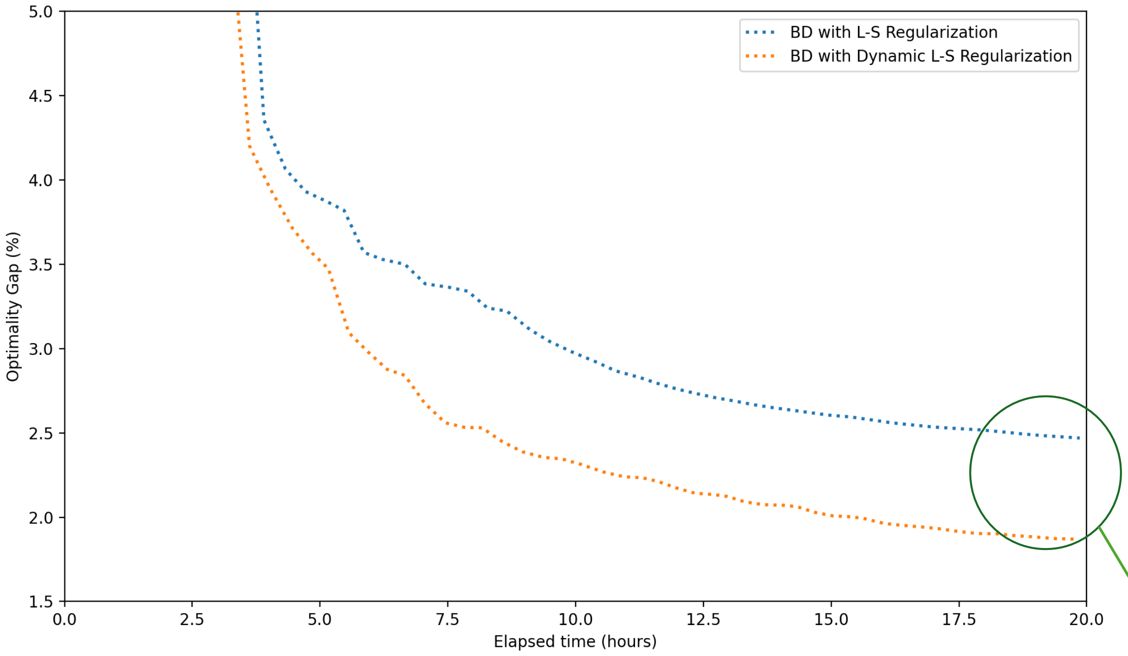
Impact of the dynamic level-set parametrization

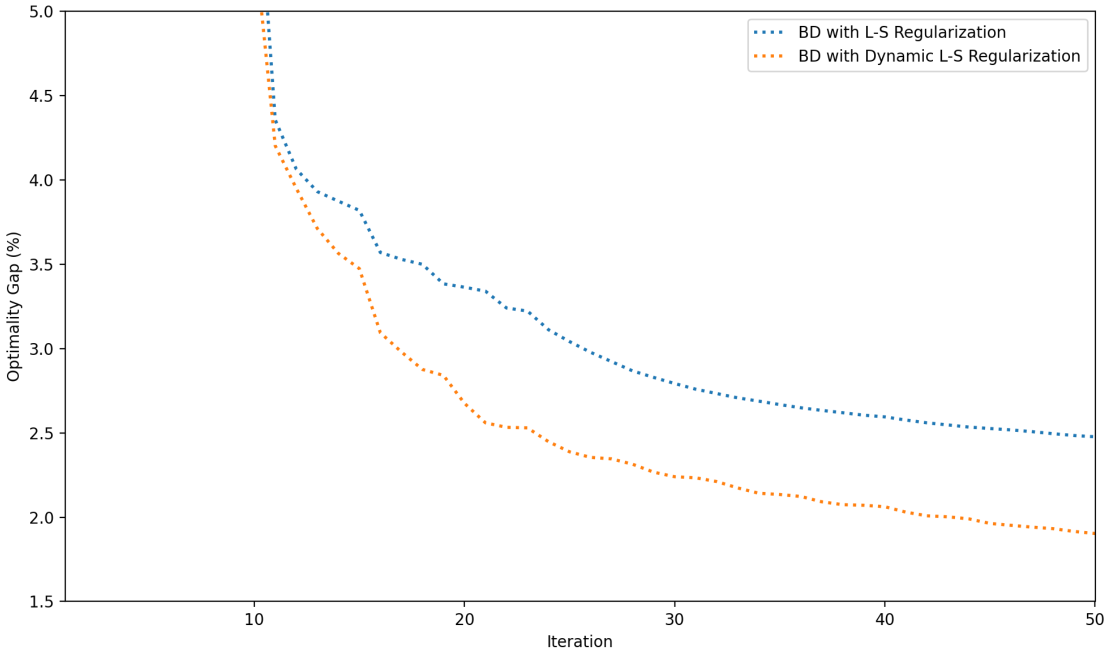




Using dynamic parametrization improves both the UB and LB

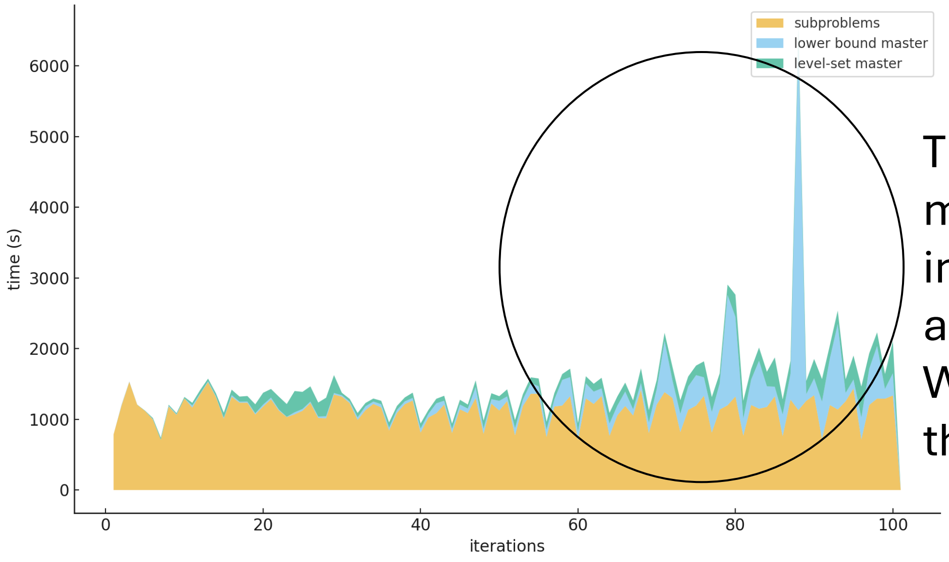
Impact of the dynamic level-set parametrization



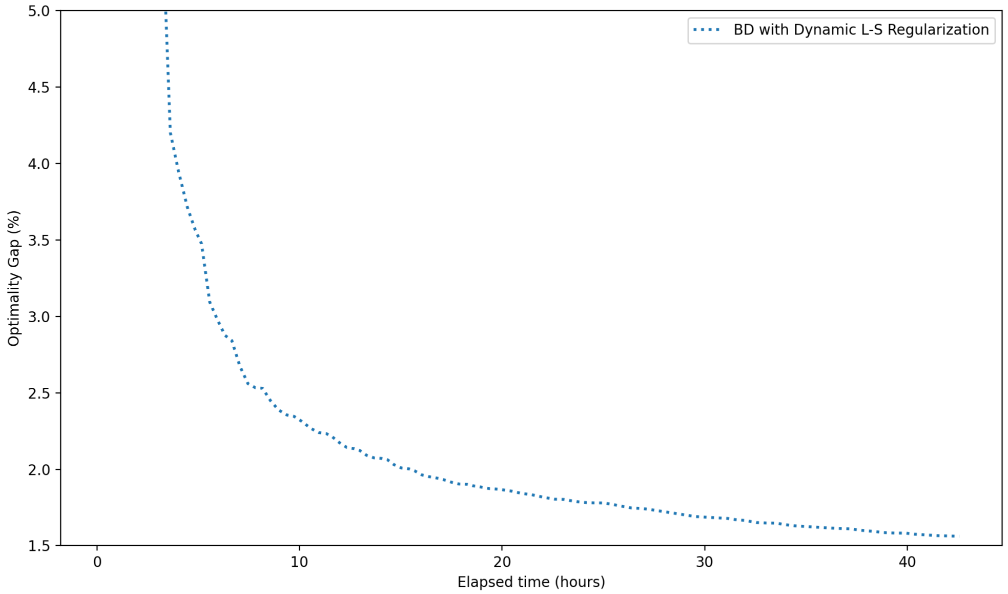


Using dynamic parametrization, the optimality gap reaches ~1.86% at 20 hours / 56 iterations, which is 0.6 % lower than the baseline (~24% relative reduction)

Key run time statistics



The Benders Decomposition master problem becomes increasingly time-consuming as iterations progress.
We therefore need to control this growth.



Using dynamic parametrization, the optimality gap reaches ~1.56% at 42 hours / 101 iterations

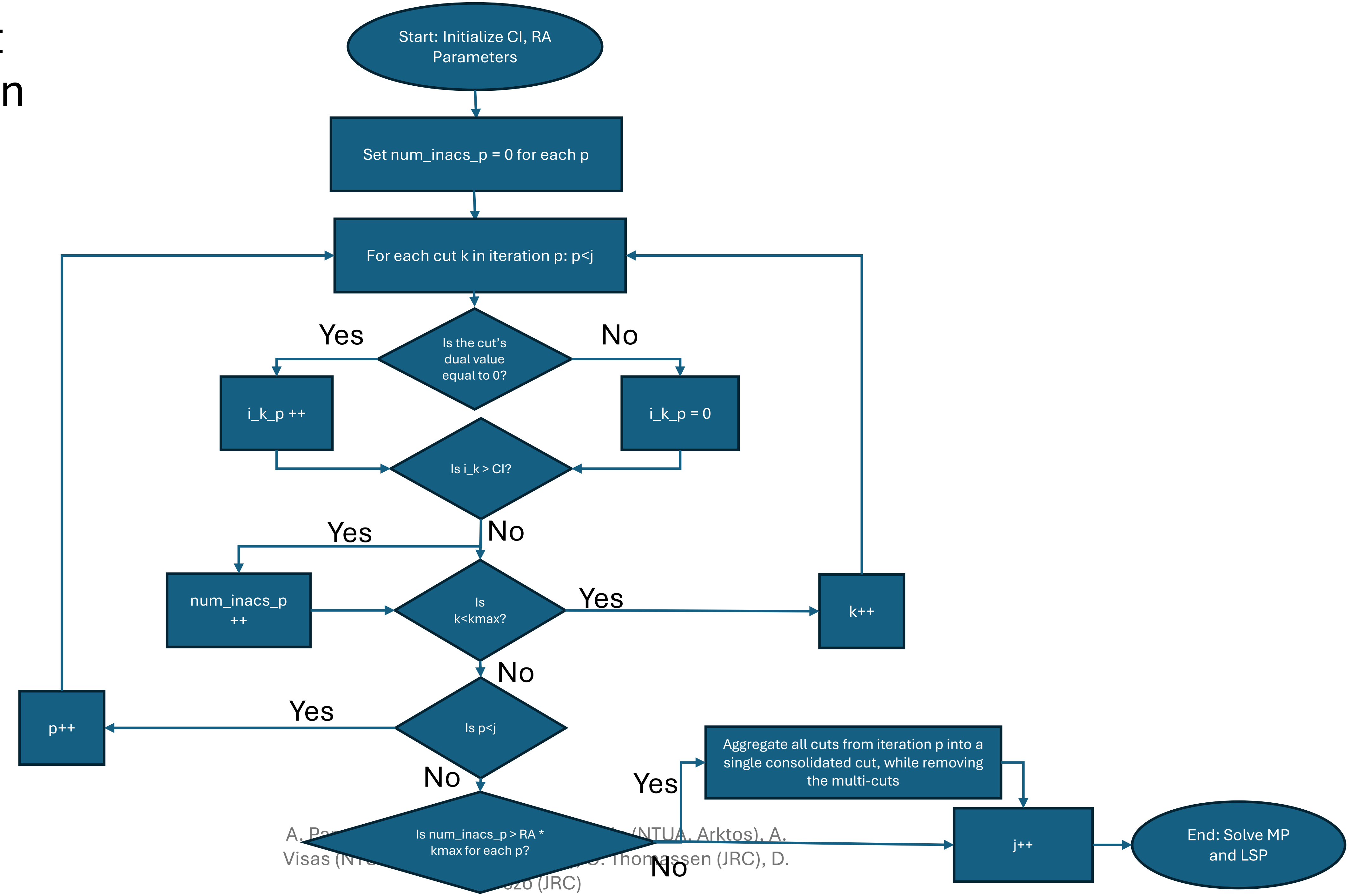
Adaptive cut consolidation [8]

- The algorithm aggregates inactive cuts from an earlier iteration to reduce memory usage and speed up the solution of both the master and level-set problems

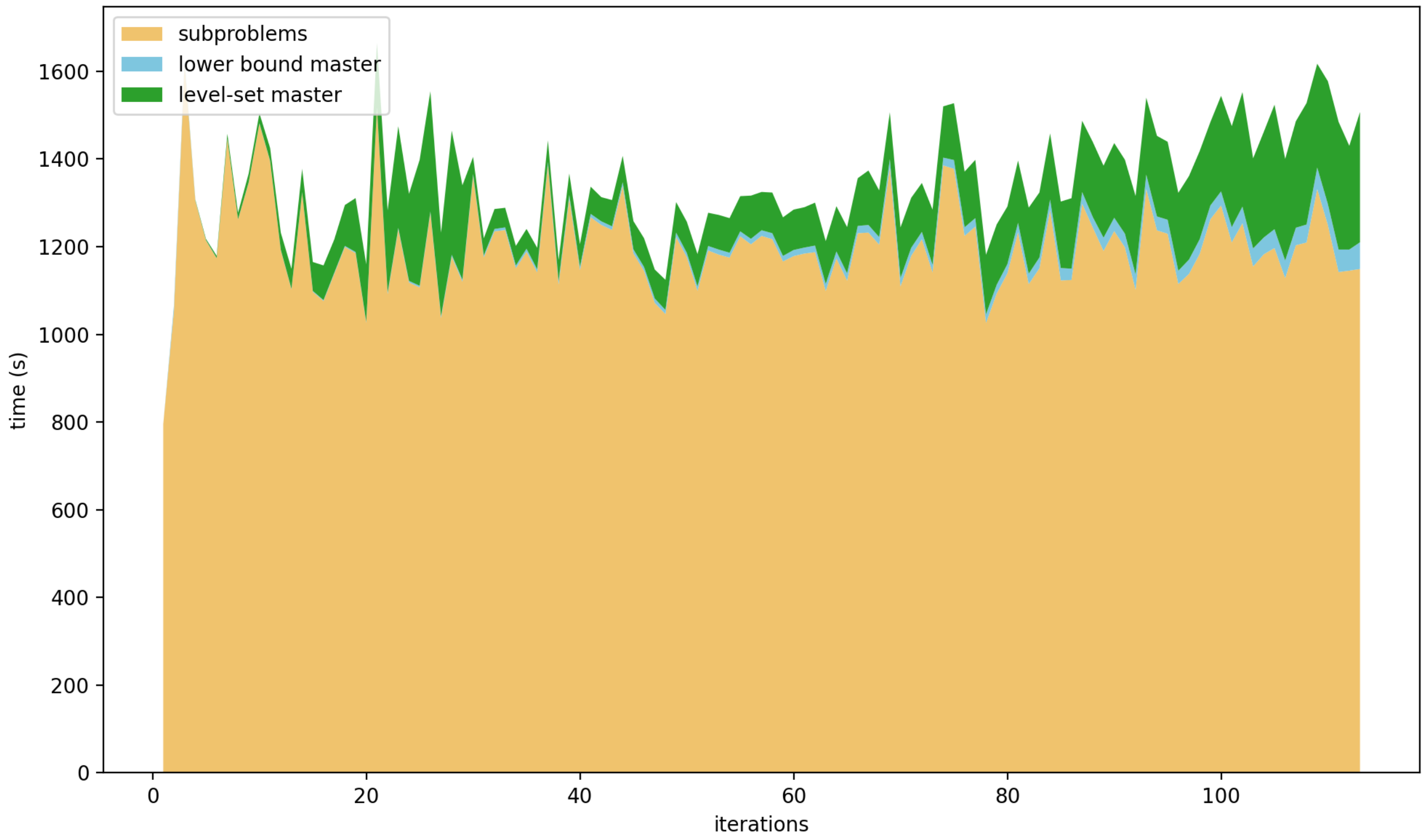
NOMENCLATURE

- CI: Consecutive Inactive Parameter (e.g. 5)
- RA: Relative Active Parameter (e.g. 0.95)
- i_k_p: counter of consecutive zero-dual instances for cut k from iteration p
- num_inacs_p: number of cuts from iteration p which are considered removable
- kmax = |scenarios|*|periods|: the maximum number of cuts created in each iteration
- j: current iteration index
- p: a previous iteration index
- k: cut index within an iteration

Adaptive cut consolidation flowchart

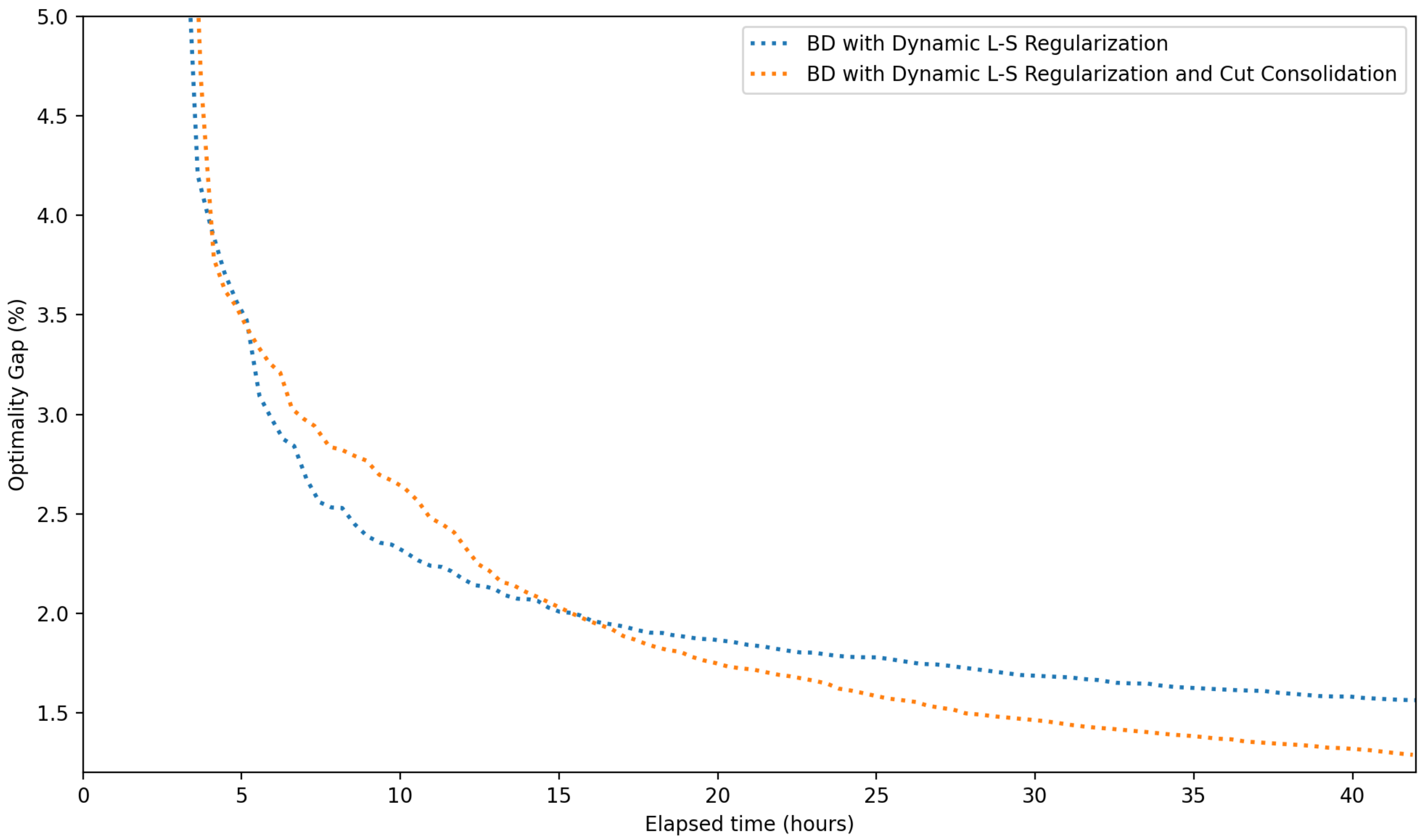


Impact of the dynamic level-set parametrization along with the adaptive cut consolidation



Cut consolidation delivered significant speedups in the master problem (and reduced spikes), whereas the level-set problem benefited, but to a smaller extent

Optimality gap comparison



Using dynamic parametrization with cut consolidation, the optimality gap reaches ~1.28% at 42 hours / 110 iterations, which is 0.28 % lower than the previous baseline (~18% relative reduction)

Conclusions

Key conclusions

- We present a decomposition scheme implemented in high performance computing infrastructure that is tailored to tackle the clearing of renewable capacity auctions at European scale
    - The algorithm has proven to be successful at regional scale (CWE, which comprises 40% of the entire EU)
    - The problem is massive, as it consists of more than a dozen uncertainty scenarios, thousands of time steps per scenario, and thousands of nodes on a power system that obeys power flow constraints
- Our decomposition scheme relies on recent innovations for temporal decoupling of storage constraints
    - Which we combine with regularization techniques and dynamic programming techniques based on value function approximation
- The decomposition scheme is well tailored to a high performance computing implementation
    - The subproblem structure allows the usage of a massive array of computational nodes and CPUs
- Results reveal a non-negligible effect of uncertainty on investment decisions

Thank you

Questions?

For more information:

https://ap-rg.eu/

papavasiliou@mail.ntua.gr

References

[1] The European Commission. Communication from the Commission: The European Green Deal, 2019

https://commission.europa.eu/publications/communication-european-green-deal_en

[2] The European Commission. Report from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions: Renewable Energy Progress Report, 2020

https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex:52020DC0952

[3] Q. Lete, Y. Smeers, A. Papavasiliou, Power Generation Investment under Zonal Electricity Pricing with Market-Based Redispatch, forthcoming in the Energy Journal

https://ap-rg.eu/wp-content/uploads/2025/01/J51.pdf

References (II)

[4] D. Avila, A. Papavasiliou, M. Junca, E. Exizidis, Applying High-Performance Computing to the European Resource Adequacy Assessment, IEEE Transactions on Power Systems, vol. 39, no. 2, pp. 3785-3797, March 2024

https://ap-rg.eu/wp-content/uploads/2023/09/J45.pdf

[5] M. Zampara, A. Visas, D. Avila, L. Kliafas, A. Papavasiliou, A Comparison of Alternative Algorithms for Resolving Stochastic Expansion Planning Problems with Endogenous Adequacy Targets, 21st International Conference on the European Energy Market, 2025

https://ap-rg.eu/wp-content/uploads/2025/05/C38.pdf

[6] Papavasiliou, Anthony. Optimization models in electricity markets. Cambridge University Press, 2024

References (III)

[7] A. Jacobson, F. Pecci, N. Sepulveda, Q. Xu, and J. Jenkins, “A computationally efficient Benders decomposition for energy systems planning problems with detailed operations and time-coupling constraints,” INFORMS Journal on Optimization, vol. 6, 2023.

[8] C. Wolf, A. Koberstein, “Dynamic sequencing and cut consolidation for the parallel hybrid-cut nested L-shaped method”, European Journal of Operational Research, 2013.

[9] Zhang, Hongyu, et al. "A stabilised Benders decomposition with adaptive oracles for large-scale stochastic programming with short-term and long-term uncertainty." *Computers & Operations Research* 167 (2024): 106665.