Supported by:
Federal Ministry for Economic Affairs and Energy
on the basis of a decision by the German Bundestag

Methods to reduce computation times of linear optimising Energy System Models

Kai von Krbek, Karl-Kiên Cao, Manuel Wetzel

DLR, Department of Energy Systems Analysis

WeatherAggReOpt, Essen (online)

2020-04-03

DLR

Content

- Motivation
- What systems were investigated and their characteristics and dimensions
- Model based speed up methods
    - Downscaling (temporal and spatial)
    - Rolling Horizon and heuristics
- Conclusion

Motivation

- Highly temporal and spatial resolved models are computationally intensive.
- How can we reduce computation times?
- If we simplify the models, how much accuracy do we lose?
- What are the best speed-up methods and how much faster do we get?

How can we speed-up energy system models (ESM)

- Efficient coding
- Solver parameterisation tuning
- Buy a bigger/better computer
- Reduce the size of the ESM

What systems were investigated

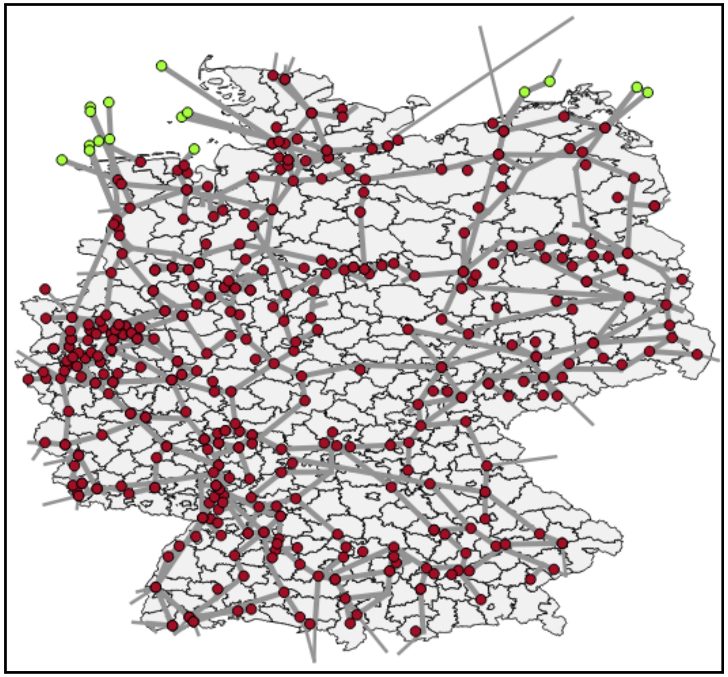
- linear optimising energy system models
    - computing times > 12h (dominated by solver)
    - storage and transmission as characteristic components of the ESM
    - dispatch and investment models
- shared memory hardware
- usage of standard solvers (e.g. CPLEX)
- formulated in GAMS

Main dimensions of ESM

- temporal dimensions
  - typical-days / typical periods
  - hours in year (8760h)
- spatial dimensions
  - NUTS 0/1/2/3
  - extra high voltage transmission network
- technological dimension
  - power plants
    - conventional power plants
    - VRE plants (hourly feed-in time series)
  - transmission
  - storage

Evaluated system

Model name	REMix
Author (Institution)	German Aerospace Center (DLR)
Model type	Linear programing minimization of total costs for system operation **economic dispatch** / optimal dc power flow with **expansion of storage and transmission capacities**
Sectoral focus	Electricity
Geographical focus	Germany
Spatial resolution	> 450 nodes (reference model)
Analyzed year (scenario)	2030
Temporal resolution	8760 time steps (hourly)



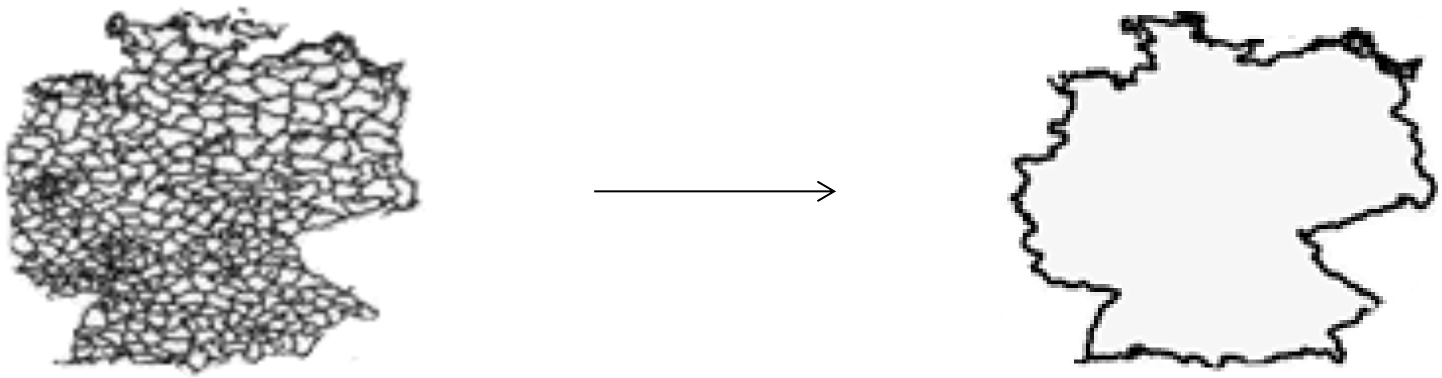
Solver	Commercial
Algorithm	Barrier
Cross-over	Disabled
Max. parallel barrier threads	16
Scaling	Aggressive

Recapture: solver based improvements and general improvements

- Reduce the accuracy of the result
  - i.e. reduce the solver tolerance
- don’t do crossover, if it is not necessary
  - this leads to results that is very close to the optimum (solver tolerance)
  - you will not get the basic feasible solution
  - caveat: there might be negative numbers in results of positive variable
    - additional effort with analysis of the results
    - if results are used for a “warm-start” of another solve, this might lead to infeasibilities

Model based speed up methods

- pure model reduction (downscaling)
  - temporal (averaging time-steps)
  - spatial
- rolling horizon and temporal zooming heuristics



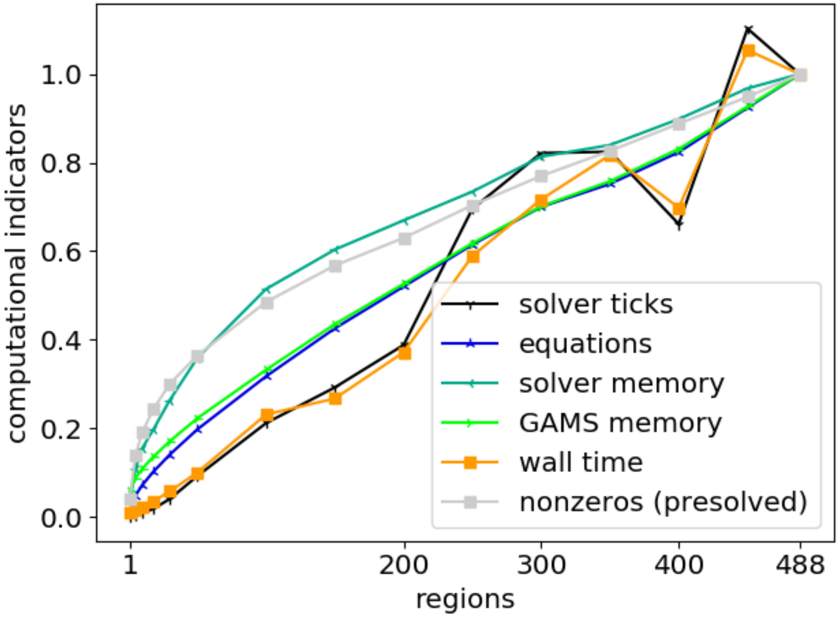
Spatial and temporal scaling

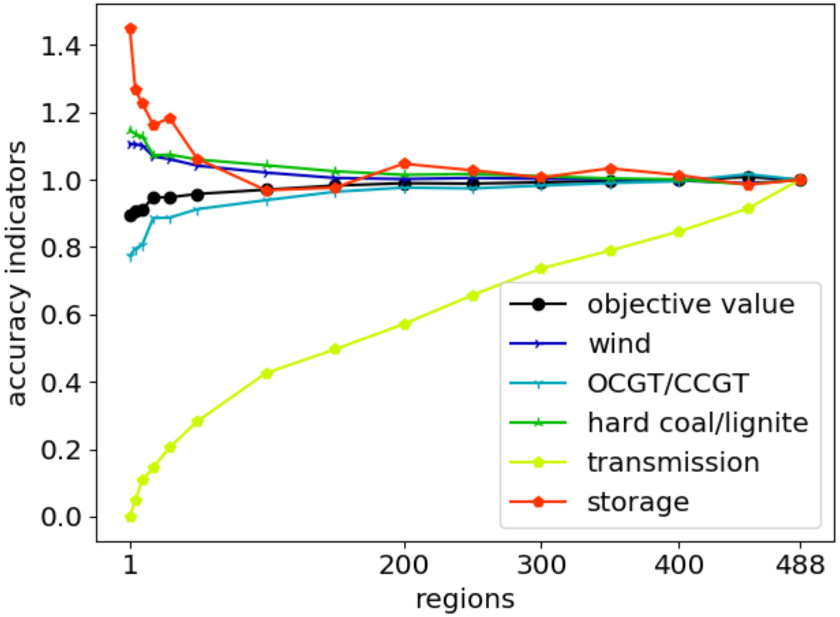
- the parameterisations will be investigated with and without expansion of transmission and storage
- spatial aggregation
  - aggregation through hierarchical clustering of the nodes
  - the transmission lines will be summed accordingly
  - within one region, a copper plate is assumed
  - at full aggregation, Germany is one big copper plate
- temporal aggregation
  - temporal averaging of feed-in time series and demand

Spatial scaling without expansion

CPLEX ticks 16.3 Mio.
Total memory 79 GB
GAMS time 0.6 h
Total wall-clock time 3.6 h

Objective 21.9 Bio €
Wind 162 TWh
Gas 174 TWh; Coal 105 TWh
Storage 4.1 TWh
Transmission 434 TWh



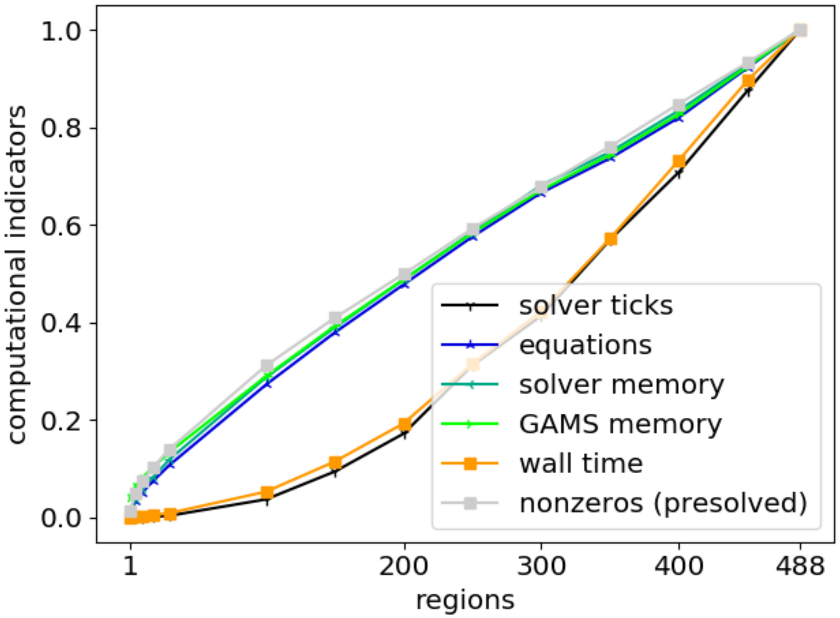


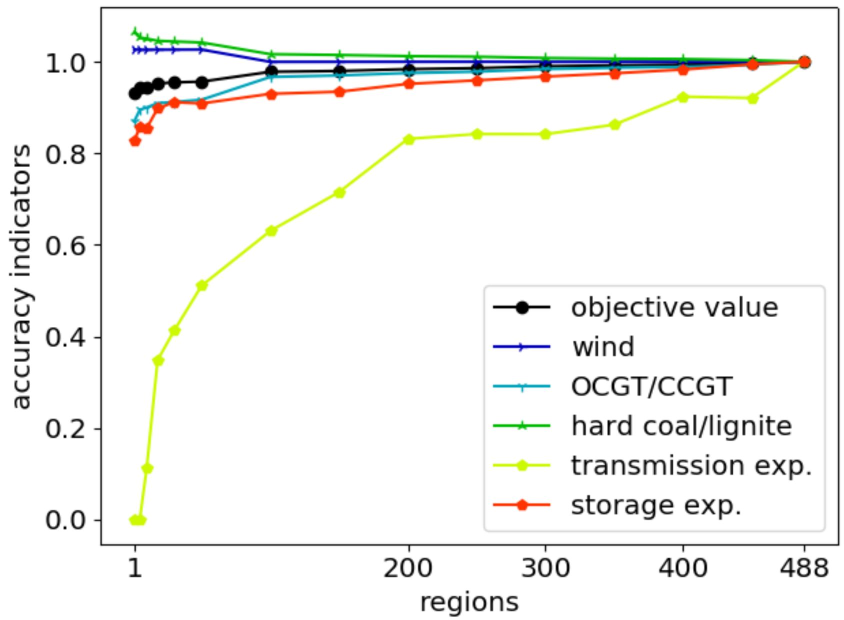
DLR

Spatial scaling with expansion

CPLEX ticks 381.3 Mio.
Total memory <256 GB
GAMS time 6.6 h
Total wall clock time 50.9 h

Objective 23.2 Bio €
Wind 175 TWh
Gas153 TWh; Coal 115 TWh
Storage expansion 123 GWh; Transmission expansion 28.8 GW

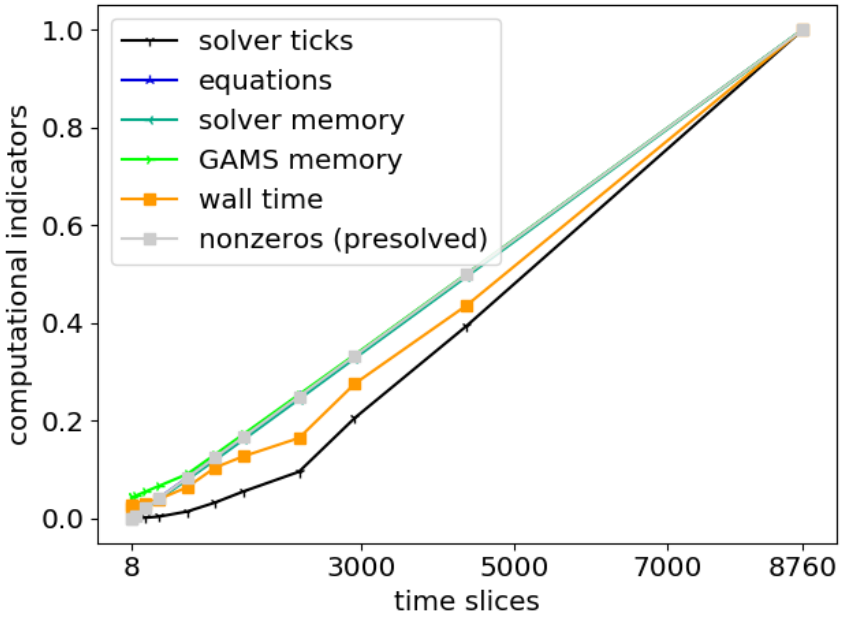


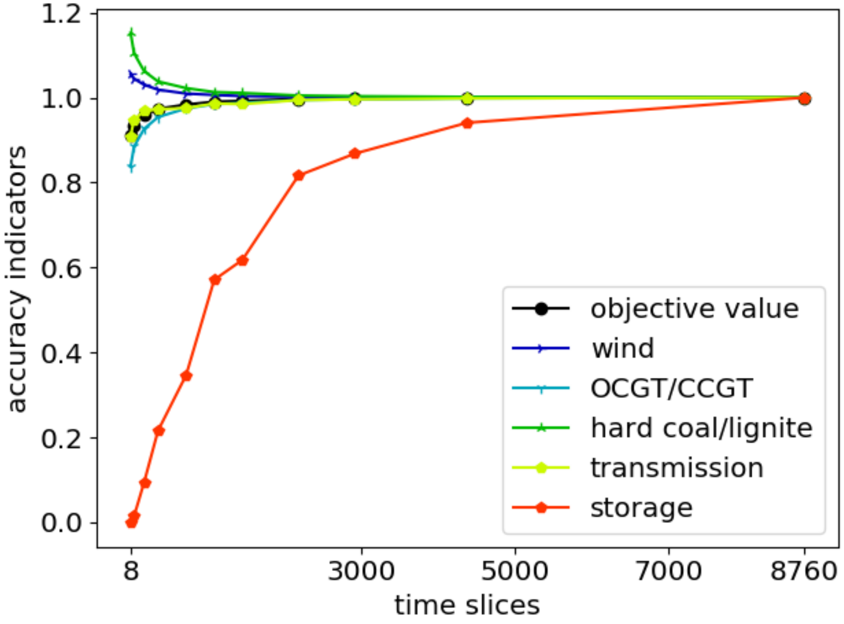


Temporal scaling without expansion

CPLEX ticks 16.3 Mio.
Total memory 79 GB
GAMS time 0.6 h
Total wall-clock time 3.6 h

Objective 21.9 Bio €
Wind 162 TWh
Gas 174 TWh; Coal 105 TWh
Storage 4.1 TWh
Transmission 434 TWh

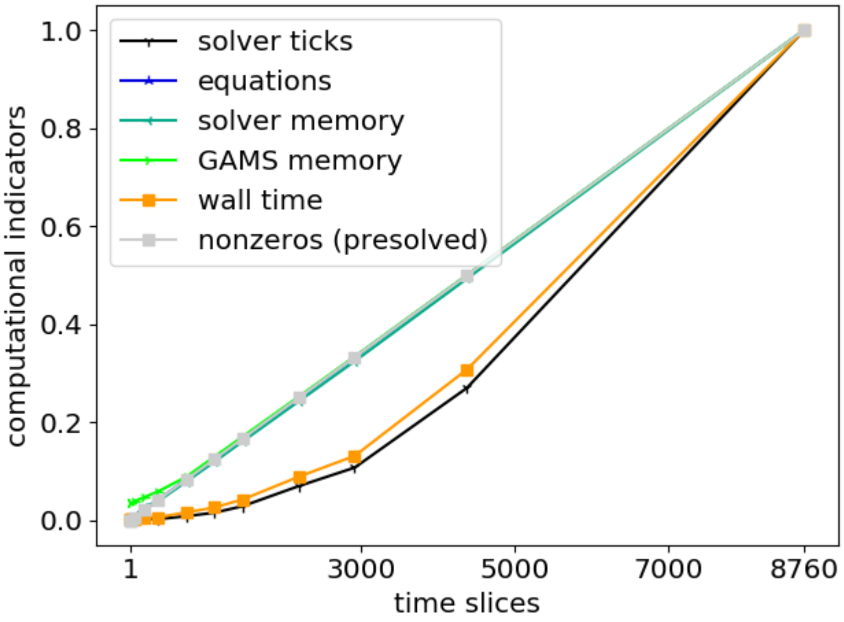


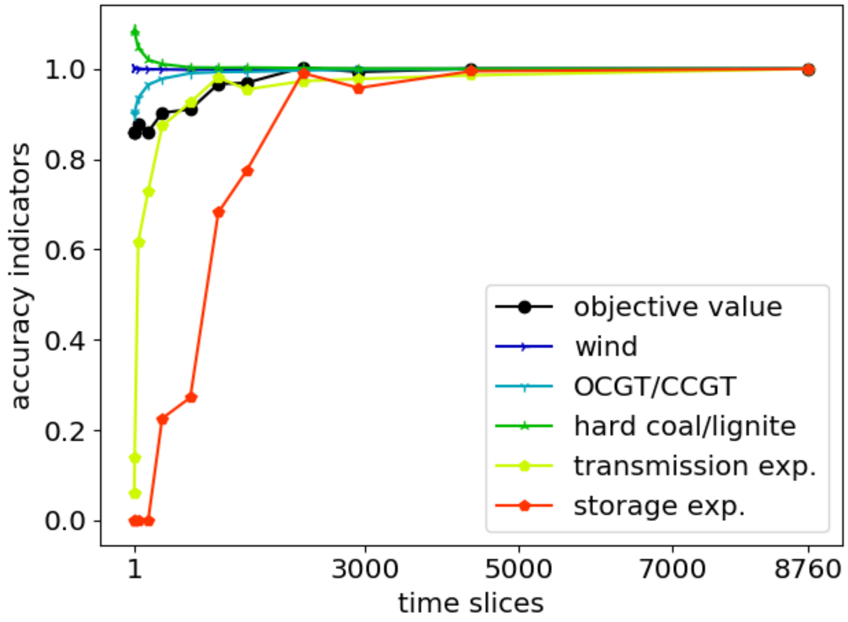


Temporal scaling with expansion

CPLEX ticks 16.3 Mio.
Total memory 79 GB
GAMS time 0.6 h
Total wall-clock time 3.6 h

Objective 21.9 Bio €
Wind 162 TWh
Gas 174 TWh; Coal 105 TWh
Storage 4.1 TWh
Transmission 434 TWh





Speed up

spatial

- transmission has a strong deviation
- without expansion
    - speed-up 4. accuracy ~70-95%
- with expansion
    - speed-up 8. accuracy ~70-95%

temporal

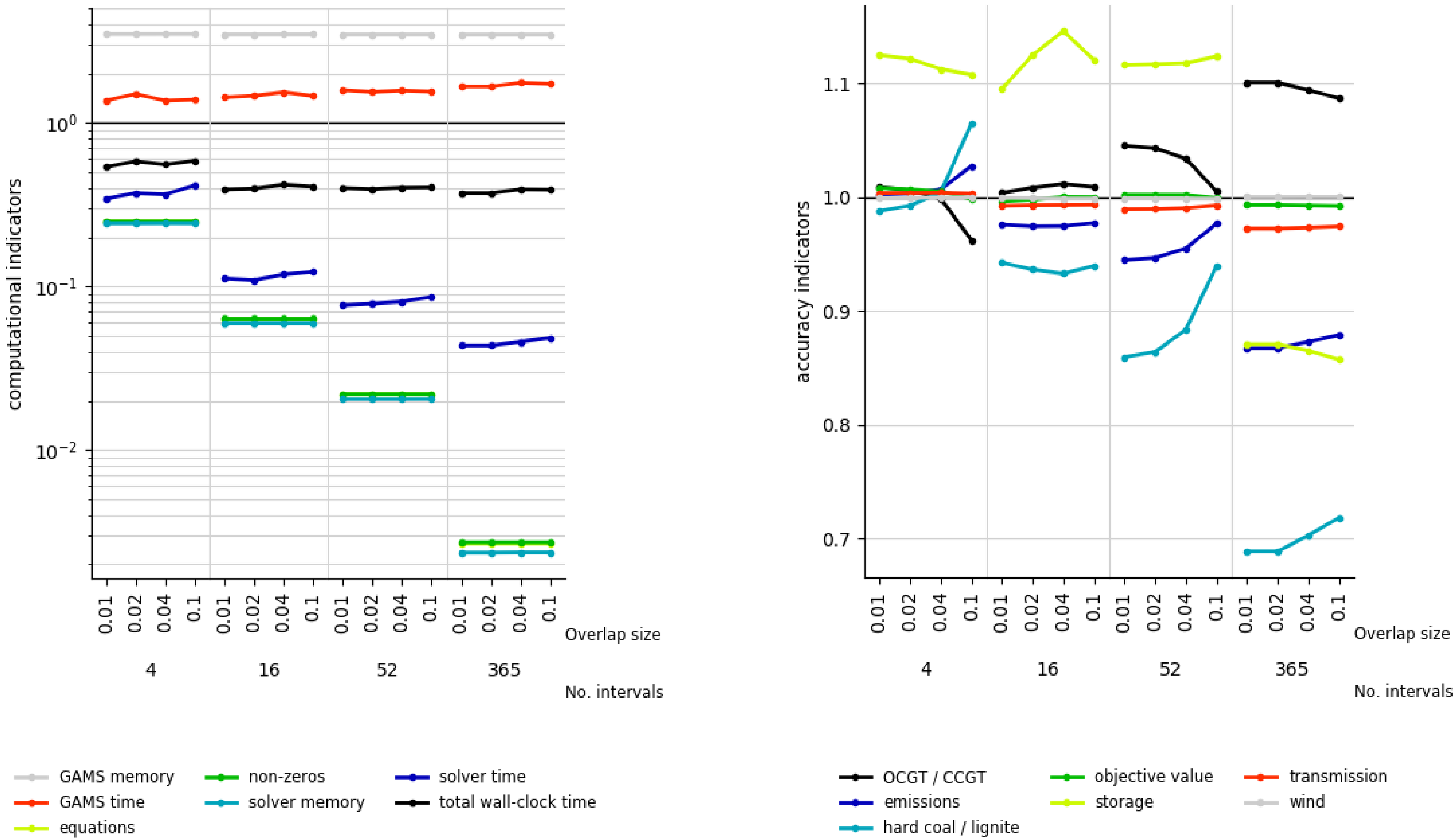
- storage has a strong deviation
- without expansion
    - speed-up 6. accuracy ~80-95%
- with expansion
    - speed-up 10. accuracy ~85-95%

- The larger the system is, the greater the speed-up will be.
- For small systems the speed-up might be negligible.

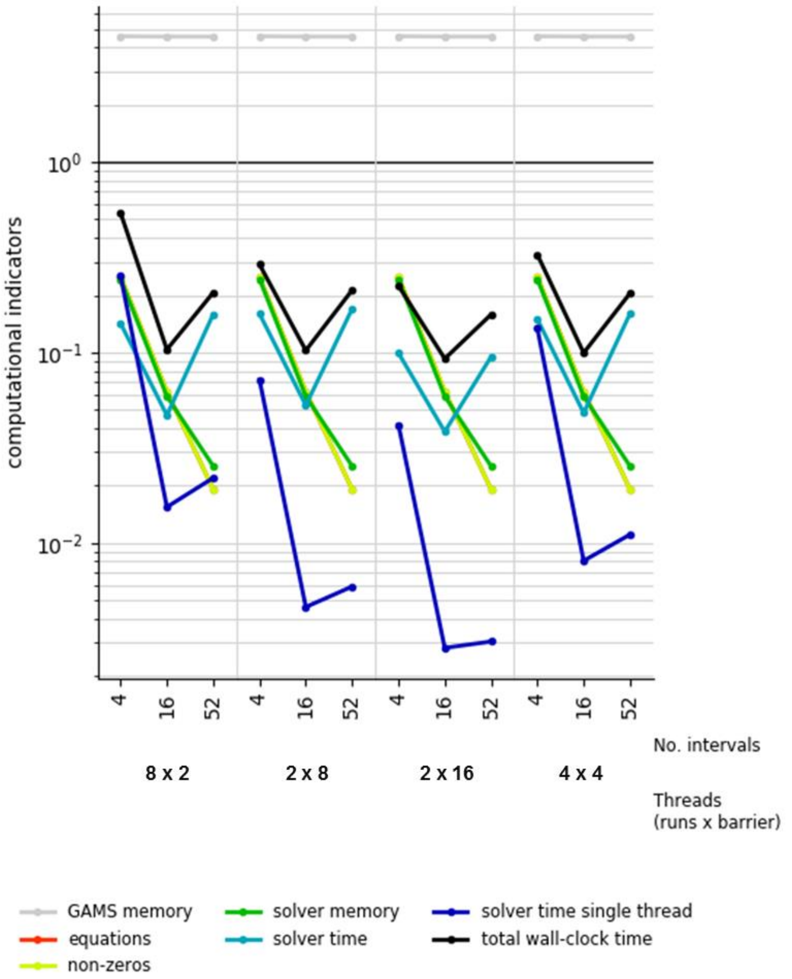
Rolling horizon

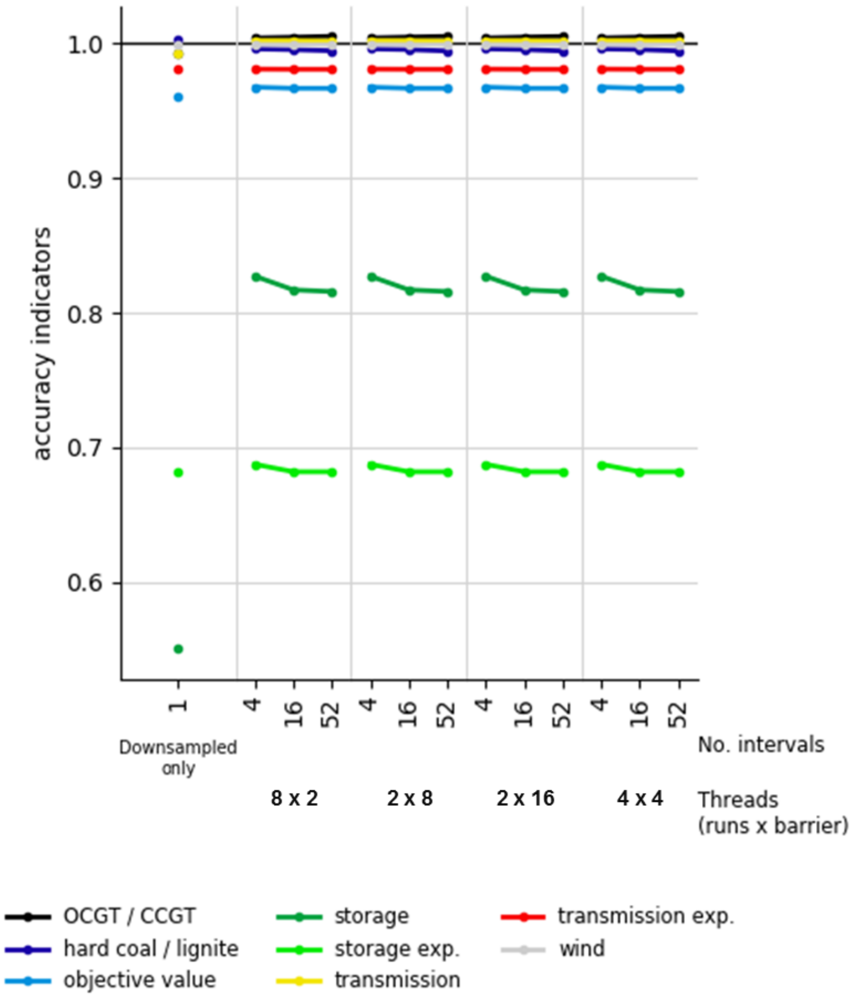
- Implemented on the temporal scale
- Pure rolling horizon
  - only dispatch, no expansion planning
  - variable overlap of the intervals for more realistic boundary conditions
  - with higher number of interval the accuracy gets worse
- Temporal zooming heuristics
  - low resolution (8 hourly) initial run to fix the boundary conditions (b.c.)
  - sub-intervalls in the year will be calculated with 8h-run as b.c.
  - expansion possible

Rolling horizon



Temporal zooming heuristics





Rolling Horizon - Summary

- Speed-up of factor five
- Less accurate than full model (up to 30%)
- storage usage is more accurate than simple down-sampling
- bigger memory usage
- better representation of actual operation => full year optimisation is not realistic

Key take-aways

- Temporal or spatial scaling work best
  - use appropriate scaling for the chosen question
    - temporal reduction works well for transmission (expansion)
    - spatial clustering works well for storage (expansion)
- Rolling horizon can improve the result performance, but uses more RAM
- First steps should be the improvement of lean formulation of the ESM

Thank you for your attention

Supported by:

Federal Ministry for Economic Affairs and Energy

on the basis of a decision by the German Bundestag

Kai von Krbek
Kai.Krbek@dlr.de
https://www.dlr.de/tt/

BEAM-ME

A PROJECT BY

GAMS
TU berlin
HLRS
JÜLICH FORSCHUNGSZENTRUM
DLR Deutsches Zentrum für Luft- und Raumfahrt German Aerospace Center
ZIB

DLR

Recapture: source code improvement (example GAMS)

- input data should not differ much in order of magnitude
- use adequate precission of input data
- use of „option kill“, e.g. for long time-series input data, saves memory
- abundant use of contraints (dolar condition) over domain of defintion
- avoidance of consideration of technologies providing power at same costs
  - i.e. no decisions of equal options

- Helpful reference for GAMS:
  “Speeding up GAMS Execution Time” by Bruce A. McCarl
  https://www.gams.com/mccarl/speed.pdf