BEAM-ME: Accelerating Linear Energy Systems Models by a Massively Parallel Interior Point Method

T. Breuer, M. Bussieck, K.-K. Cao, F. Fiand, H. C. Gils, A. Gleixner, D. Khabi, N. Kempke, T. Koch, D. Rehfeldt, M. Wetzel

published in

NIC Symposium 2020
M. Müller, K. Binder, A. Trautmann (Editors)

Forschungszentrum Jülich GmbH,
John von Neumann Institute for Computing (NIC),
Schriften des Forschungszentrums Jülich, NIC Series, Vol. 50,
ISBN 978-3-95806-443-0, pp. 345.
http://hdl.handle.net/2128/24435

BEAM-ME: Accelerating Linear Energy Systems Models by a Massively Parallel Interior Point Method

Thomas Breuer[2], Michael Bussieck[3], Karl-Kiên Cao[4], Frederik Fiand[3], Hans Christian Gils[4], Ambros Gleixner[1], Dmitry Khabi[5], Nils Kempke[1], Thorsten Koch[1], Daniel Rehfeldt[1], and Manuel Wetzel[4]

[1] Zuse Institute Berlin/Technische Universität Berlin, Berlin, Germany
E-mail: {gleixner, kempke, koch, rehfeldt}@zib.de

[2] Jülich Supercomputing Centre (JSC), Forschungszentrum Jülich, Jülich, Germany
E-mail: t.breuer@fz-juelich.de

[3] GAMS Software GmbH, Frechen, Germany
E-mail: {mbussieck, ffiand}@gams.com

[4] Department of Energy Systems Analysis, Institute of Engineering Thermodynamics, German Aerospace Center (DLR), Stuttgart, Germany
E-mail: {karl-kien.cao, hans.gils, manuel.wetzel}@dlr.de

[5] High-Performance Computing Center Stuttgart (HLRS), Stuttgart, Germany
E-mail: khabi@hlrs.de

1 Introduction

The German energy revolution requires radical rethinking in business and science. A wide range of measures have been adopted by politicians in the course of the gradual switch to renewable energies: the last nuclear power plant is to be taken off the grid in 2022. By 2025, 40 to 45 percent of the electricity consumed nationwide will be generated from renewable energies. According to the coal commission's recommendation, the last coal-fired power plant should be shut down by 2038 at the latest. These and other changes will gradually lead to a fundamental restructuring of the energy system. Wind turbines and solar cells will be installed, power highways built and smart grids set up, energy infrastructures connected and new electricity storage technologies researched.

As concrete as this may sound, many of the factors in the planning game are uncertain. The development of the energy market is difficult to predict, and the climate outlook also raises questions. How will the sun shine in 50 years? When and where will the wind blow? Each eventuality that is fed into one of the analytical models for the energy systems of the future as a measurement, assumption or probability produces a more complex picture. This complexity is due to gigantic amounts of data that quickly push conventional computers and algorithms to the limits of their capacity and performance. The challenges of the energy system transformation are to a large extent challenges of information technology.

A widely used means in research and industry to analyse today's and future energy systems are Energy System Models (ESMs). Such models implement for instance the hourly dispatch of power stations, power transfer, and long-term investment decisions,

including sector coupling with transport and heating. As such, ESMs are key to enable studies regarding policy design, analysis of technology pathways, or the analysis of future energy systems. One of the most successful modelling and solution methods for modern ESMs is linear programming. Linear programs (LPs) can be used to formulate and compute cost-optimal states of an energy system and its extensions.

There has been tremendous progress in general LP solvers during the last 30 years.[1, 2] Many LPs that were considered intractable two or three decades ago can now be solved within seconds. Still, for large-scale LPs with hundreds of millions of variables and constraints such as those arising from high-resolution energy system models, even the best commercial solvers can take prohibitively long to find an optimal solution. Moreover, such large-scale problems might not even fit into the main memory of a modern desktop machine. On the other hand, the increasing availability of distributed-memory parallel computers offers a huge potential both for reducing solution time and avoiding memory bottlenecks. Unfortunately, general state-of-the-art LP solvers cannot (efficiently) run on distributed-memory systems. Therefore, one way forward is to develop more specialised algorithms for a distributed parallel solution, exploiting structures commonly found in ESMs.

Against this backdrop, the BEAM-ME project was initiated: an interdisciplinary research project carried out between 2015 and 2019 as part of the sixth energy research program of the German federal government (funding code: 03ET4023A-F). The participants were the Energy Systems Analysis Department of the German Aerospace Center, the Jülich Supercomputing Centre at Forschungszentrum Jülich, the High-Performance Computing Centre Stuttgart, the Mathematical Optimisation and Scientific Information Department at Zuse Institute Berlin, the Institute of Mathematics at Technische Universität Berlin and GAMS Software GmbH. This article describes the development and implementation of massively parallel algorithms for linear ESMs within the BEAM-ME project. Also, computational results for several large-scale ESMs will be presented.

2 Exhibiting Block Structure in Energy Systems Models

A typical structure observed in linear energy system models is the so-called *general arrowhead* or *doubly bordered block-diagonal* form:

$$
\begin{aligned}
\min \quad & \sum_{i=0}^{N} c_i^T x_i \\
\text{s.t.} \quad & T_0 x_0 && && && && = h_0 && (1)\\
& T_1 x_0 && + W_1 x_1 && && && = h_1 && (2)\\
& T_2 x_0 && + && W_2 x_2 && && = h_2 && (3)\\
& \vdots && && && \ddots && \vdots \\
& T_N x_0 && + && && && W_N x_N && = h_N && (4)\\
& U_0 x_0 && + U_1 x_1 && + U_2 x_2 && \cdots && U_N x_N && = h_{N+1} && (5)\\
& x_0, && x_1, && x_2, && \cdots && x_N && \geq 0 && (6)
\end{aligned}
$$

where $c_i, x_i \in \mathbb{R}^{n_i}, T_i \in \mathbb{R}^{m_i \times n_0}, U_i \in \mathbb{R}^{m_{N+1} \times n_i}$ for $i \in \{0, 1, \ldots, N\}$ and $W_i \in \mathbb{R}^{m_i \times n_i}$ for $i \in \{1, \ldots, N\}$, with $n_i, m_i \in \mathbb{N}_0$ for $i \in \{0, 1, \ldots, N\}$ and $m_{N+1} \in \mathbb{N}_0$. Note that also inequalities and variables bounds are allowed, but are not presented here for the sake of simplicity. This LP structure is quite general and can also be found in many applications beyond energy system modelling. Often, in an ESM each $i \in N$ corresponds to a point of time in a (discretised) time horizon. The *linking variables* x_0 may describe investment decisions. The *linking constraints* Eq. 5 usually include energy storage between two consecutive time steps.

For a typical ESM, the number of blocks that can be defined in the model depends on the motivation behind the research question. If, for example, a calendar year is described using time steps, a structure with 365 blocks based on the number of days can parallelise the solution process. Or exactly twice as many blocks are defined and thus days and nights are mapped separately. Not every block structure is equally suitable for the underlying algorithms. Therefore, there are different application variants. The modelling language GAMS was extended during the BEAM-ME project in such a way that the modeller can easily determine the block structure(s) of existing models by annotating the decision variables. Fig. 1 visualises several such annotations for small-scale problems.

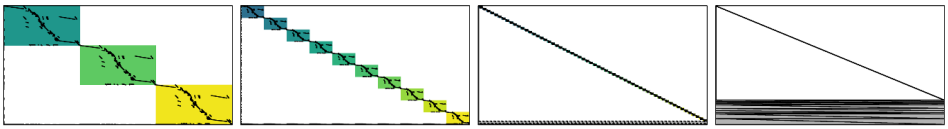

Figure 1. Non-zero plots of different block diagonal structures of a model instance. A challenge lies in finding an annotation that defines a sufficient number of blocks for parallelisation without the gray blocks of the linking variables and constraints becoming dominant.

3 A Parallel Interior Point Method for Linear Programs with General Arrowhead Form

For general LPs the two major algorithmic classes are simplex and interior-point methods.[3] Interior-point methods are often more successful for large problems, and they offer more potential for parallelisation since the main computational effort usually goes into factorising matrices. For a particular class of arrowhead LPs, two-stage stochastic optimisation problems, there exist already several distributed algorithms.[4–6] Two-stage stochastic optimisation problems do not have linking constraints and the diagonal blocks are intrinsically related. Both assumptions do not hold for typical ESMs. Moreover, a common feature of ESMs is a huge amount of linking constraints and variables (often more than 100 000). This large number of linking constraints and variables renders straightforward extensions of previous work on stochastic optimisation problems prohibitive.

To allow for the parallel solution of large-scale ESMs, we have developed and implemented several algorithms that are embedded within an interior-point method. Roughly speaking, an interior point method iteratively moves towards an optimal solution to the

given LP. The major part of the parallelisation takes place within the solution of the linear system, arising in each step of the interior-point algorithm. By suitable projection and permutation this linear system can be reduced to the following form:

$$\begin{bmatrix} K_1 & & & B_1 \\ & \ddots & & \vdots \\ & & K_N & B_N \\ B_1^T & \cdots & B_N^T & K_0 \end{bmatrix} \begin{bmatrix} \Delta z_1 \\ \vdots \\ \Delta z_N \\ \Delta z_0 \end{bmatrix} = \begin{bmatrix} b_1 \\ \vdots \\ b_N \\ b_0 \end{bmatrix} \tag{7}$$

where

$$K_i = \begin{bmatrix} D_i & W_i^T \\ W_i & 0 \end{bmatrix},\ K_0 = \begin{bmatrix} D_0 & T_0^T & U_0^T \\ T_0 & 0 & 0 \\ U_0 & 0 & 0 \end{bmatrix},\ B_i = \begin{bmatrix} 0 & 0 & U_i^T \\ T_i & 0 & 0 \end{bmatrix} \tag{8}$$

The parallelisation is realised as specialised Schur complement decomposition, consisting of the following steps:

1. Multiply each row $i = 1, ..., N$ of Eq. 7 by $-B_i^T K_i^{-1}$.
2. Sum up all rows.
3. Solve $\left(K_0 - \sum_{i=1}^{N} B_i^T K_i^{-1} B_i\right) \Delta z_0 = b_0 - \sum_{i=1}^{N} B_i^T K_i^{-1} b_i$.
4. For each row $i = 1, ..., N$ insert Δz_0 and compute Δz_i.

The matrix

$$C = K_0 - \sum_{i=1}^{N} B_i^T K_i^{-1} B_i$$

is called the Schur complement of the above system. While most of the above steps can be parallelised efficiently, the formation of the actual Schur complement and the solution of the resulting linear system form a bottleneck and we have put great effort into improving this part of the algorithm.

The implementation builds on the existing solver PIPS-IPM,[7, 8] a solver for two-stage stochastic quadratic programs. The first step was the extension of this solver to handle linking constraints. Also, an interface to GAMS was implemented. However, a large number of further changes were necessary, to handle for example the strong linkage between individual blocks in typical ESM LPs, or the bad condition of these LPs. Also, considerable work had to be put into bringing the sequential run time closer to that of the leading commercial LP solvers (which have been developed over decades by the leading experts in the field). Most important among the newly implemented (and designed) methods are

- multiple-corrector interior-point algorithms,
- parallel LP scaling methods,
- parallel presolving methods,

- parallel preconditioning of the Schur complement.

More details on the preconditioned Schur complement approach can be found in Ref. 9. The parallel presolving algorithms are described in Ref. 10. Overall, more than 60 000 lines of PIPS-IPM code have been modified or added, which constitutes more than half of the entire active code base of the solver. Because of several requests from the energy system community, the new solver will be made publicly available at the end of 2019. In the remainder of this article, we will refer to the new solver as PIPS-IPM++, which is also the current working name.

4 Numerical Results

The numerical results reported in this section were obtained on the supercomputer JUWELS at the Jülich Supercomputer Centre. JUWELS is equipped with 2271 standard compute nodes (Dual Intel Xeon Platinum 8168), each with 96 GB memory and 2x24 cores at 2.7 GHz. JUWELS also includes 240 large memory compute nodes with 192 GB each (and otherwise the same configuration). The nodes are connected via a Mellanox EDR InfiniBand high-speed network. Only the standard compute nodes were used for PIPS-IPM++.

Within the BEAM-ME project modelling experts from GAMS together with energy system modellers from the German Aerospace Center developed a simplified, but representative, energy system model, called SIMPLE. While SIMPLE lacks many details that are considered in full-fledged energy system models, it is easily adaptable in size (*e. g.* the number of variables or diagonal blocks) and thus highly useful for testing new solution approaches. Fig. 2 shows the scaling behaviour of PIPS-IPM++ on a relatively small SIMPLE instance (5 144 806 constraints and 5 681 482 variables) on up to 512 MPI processes. Furthermore, the figure shows the scaling behaviour of a reduced version (denoted by *New solver reduced*) of PIPS-IPM++ that just uses the Schur complement decomposition described in Sec. 3, but none of the additionally implemented algorithms such as (distributed) preconditioned, iterative solution, LP scaling, and presolving. For each MPI configuration of PIPS-IPM++ (including the reduced version) two OpenMP threads were used. The figure also shows results for the leading four commercial LP solvers CPLEX 12.8[a], Gurobi 8.1[b], MOSEK 8.1[c], and Xpress 8.4.7[d] – which can only run on shared-memory. Fig. 2 shows that the commercial solvers using one thread are considerably faster than PIPS-IPM++ using one MPI process (and two OpenMP threads) – by a factor of 5 or more. However, the commercial solvers do not exhibit a good scaling behaviour. Three of the commercial solvers show a moderate speed-up up until 12 threads, but with more threads the run-time even deteriorates. PIPS-IPM++ on the other hand scales reasonably well, and requires merely 23 seconds with the maximum number of 512 MPI processes – whereas no commercial solver achieves a run-time below 400 seconds. Furthermore, the results of the reduced version of the new solver demonstrate that the straightforward application

[a] `https://www.ibm.com/products/ilog-cplex-optimization-studio`
[b] `https://www.gurobi.com`
[c] `https://www.mosek.com/`
[d] `https://www.fico.com/en/products/fico-xpress-optimization`

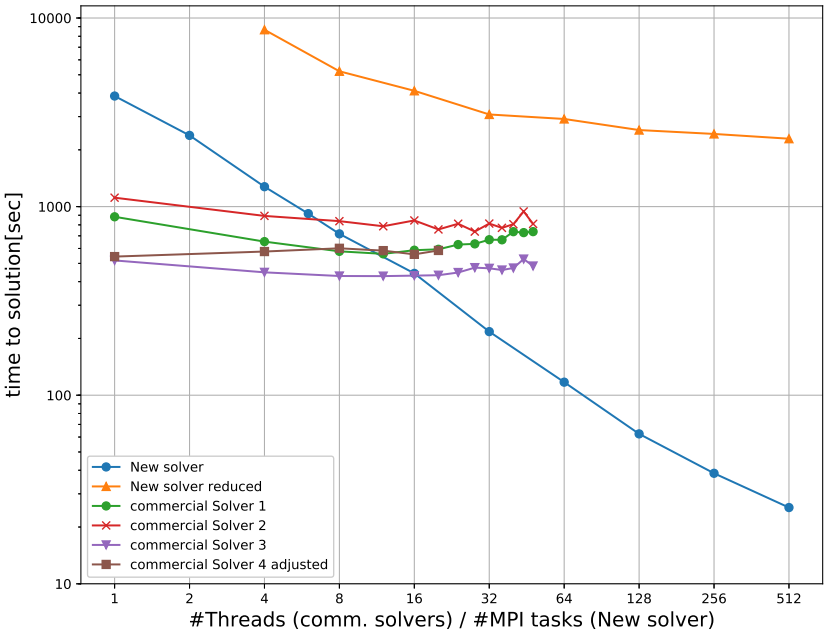

Figure 2. Scaling results of the leading commercial LP solvers (*Solver 1*,...,*Solver 4*) and PIPS-IPM++ (*new solver*) on a SIMPLE instance. PIPS-IPM++ was run with 2 OpenMP threads per MPI process.

of the Schur complement decomposition without any additional techniques is not competitive. Another notable observation is that PIPS-IPM++ is for this instance already on a shared-memory architecture highly competitive with the commercial solvers: On 32 cores (16 MPI processes, with two OpenMP threads each), PIPS-IPM++ is as fast as any commercial solver – even if one looks at their best performance among all number of threads used.

Further results are given in the following for the energy system models Premix[11] and ELMOD.[12] REMix has been developed (primarily) for the analysis of long-term energy scenarios with high shares of variable renewable energies. The ELMOD ESMs that we consider are used to analyse the impact of growing shares of renewable energies on the operation of the European transmission grid and the dispatch of conventional power plants. The first two ELMOD instances (CWE) comprise the entire transmission grid of the Central Western European region. For the other two instances (EU) this configuration was further extended to a set of 19 European countries with detailed transmission grid representation. This is the largest configuration currently possible for the ELMOD model and no instance of this configuration could previously be solved. For comparison we use the four state-of-the-art (commercial) LP solvers detailed above. We use 16 threads for each

of the commercial solvers – with more threads usually a performance degradation occurs. Note that none of the commercial solvers allow for the distributed parallel solution of LPs. Tab. 1 shows results on JUWELS of PIPS-IPM++ (with 2 OpenMP threads per MPI process) and the respective best commercial solver. ML signifies that the commercial solvers ran out of memory, even on the large nodes of JUWELS. As to the instances solvable on JUWELS, one notes that the run time per computing resources of PIPS-IPM++ is better than the best result of the commercial solvers (19 compute nodes versus 1 compute node).

	Size			PIPS-IPM++ resources		Run time (seconds)	
Instance	Variables	Constraints	Non-zeroes	MPI processes	Nodes	PIPS-IPM++	Best commercial
YSSP_exp	107 555 441	93 902 968	286 477 612	250	25	2311	ML
YSSP_disp	94 965 730	81 314 846	225 032 421	350	35	931	ML
ELMOD_CWE15	85 646 554	98 646 274	271 875 064	438	19	181	6 201
ELMOD_CWE16	85 883 074	98 909 074	272 602 144	438	19	216	6 111
ELMOD_EU15	224 677 686	254 304 961	712 452 541	876	38	1 245	ML
ELMOD_EU16	226 061 766	256 284 723	717 436 984	876	38	1 119	ML
SIMPLE_LARGE	227 060 381	206 036 266	818 449 005	1024	64	546	ML

Table 1. Computational results for large-scale instances.

5 Conclusion

The overall goal of the BEAM-ME project was to overcome the current performance limitations of linear energy systems models by exploiting high-performance computing resources. To achieve this goal on a broad scale, it was necessary to identify a common property that could be exploited algorithmically and is shared by a large majority of ESMs: block structure in the constraint matrix. The presence of this structure made it possible to develop a distributed-memory interior-point solver that showed good scalability for ESMs from different sources. This result stood out particularly in comparison to the poor scalability of today's state-of-the-art shared-memory solvers.

There are several paths for further development. One is the implementation of additional presolving methods. To improve robustness of the algorithm, we also plan to implement the homogeneous self-dual interior-point method.[13] Yet another extension, that is currently being implemented, is a hierarchical approach, which splits the Schur complement decomposition (and thus also the Schur complement) in several layers – with the aim to handle energy system models with even stronger linkage than the ones considered here.

Although HPC resources are certainly not ubiquitous, we hope that this generality and usability will help transfer the algorithmic progress achieved during the BEAM-ME project into the regular practice of the energy systems community.

Acknowledgements

The described research activities were funded by the German Federal Ministry for Economic Affairs and Energy (BMWi) within the project BEAM-ME (grant number 03ET4023A-F). The authors gratefully acknowledge the Gauss Centre for Supercomputing

e.V. (www.gauss-centre.eu) for funding this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC).

References

1. R. E. Bixby, *Solving Real-World Linear Programs: A Decade and More of Progress*, Operations Research **50**, 3–15, 2002.
2. J. Gondzio, *Interior point methods 25 years later*, European Journal of Operational Research **218**, 587–601, 2012.
3. R. J. Vanderbei, *Linear Programming: Foundations and Extensions*, International Series in Operations Research & Management Science **196**, Springer, Boston, 2014.
4. J. Barnett, J.-P. Watson, and D. L. Woodruff, *BBPH: Using progressive hedging within branch and bound to solve multi-stage stochastic mixed integer programs*, Oper. Res. Lett. **45**, 34–39, 2017.
5. M. Colombo, J. Gondzio, and A. Grothey, *A warm-start approach for large-scale stochastic linear programs*, Math. Program. **127**, 371–397, 2011.
6. C. G. Petra, O. Schenk, and M. Anitescu, *Real-Time Stochastic Optimization of Complex Energy Systems on High-Performance Computers*, Computing in Science & Engineering **16**, 32–42, 2014.
7. M. Lubin, C. G. Petra, M. Anitescu, and V. Zavala, *Scalable stochastic optimization of complex energy systems*, in Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis (SC'11), Article No. 64, 2011.
8. C. G. Petra, O. Schenk, M. Lubin, and K. Gärtner, *An Augmented Incomplete Factorization Approach for Computing the Schur Complement in Stochastic Optimization*, SIAM J. Scientific Computing **36**, C139–C162, 2014.
9. D. Rehfeldt, H. Hobbie, D. Schönheit, A. Gleixner, T. Koch, and D. Möst, *A massively parallel interior-point solver for linear energy system models with block structure*, Zuse Institute Berlin (ZIB) Rep. **19-41**, 2019.
10. A. Gleixner, N.-C. Kempke, T. Koch, D. Rehfeldt, and S. Uslu, *First Experiments with Structure-Aware Presolving for a Parallel Interior-Point Method*, in Operations Research Proceedings, under review, 2019.
11. K. Cao, J. Metzdorf, and S. Birbalta, *Incorporating Power Transmission Bottlenecks into Aggregated Energy System Models*, Sustainability **10**, 1–32, 2018.
12. F. Hinz and D. Möst, *Techno-Economic Evaluation of 110 kV Grid Reactive Power Support for the Transmission Grid*, IEEE Transactions on Power Systems **33**, 4809–4818, 2018.
13. Robert J. Vanderbei, *The Homogeneous Self-Dual Method*, in Linear Programming, International Series in Operations Research & Management Science **114**, Springer, Boston, 361–381, 2008.