A Tutorial for Using an Open-Source Solver for the Regional Energy Deployment System (ReEDS) Model

Kelly Eurek, Madeline Macmillan, and Wesley Cole

June 2021

Overview

- **Purpose:** Provide context and understanding for the implications of using an open-source solver for the Regional Energy Deployment System (ReEDS) model. We use the ReEDS model version 2020 for this tutorial.

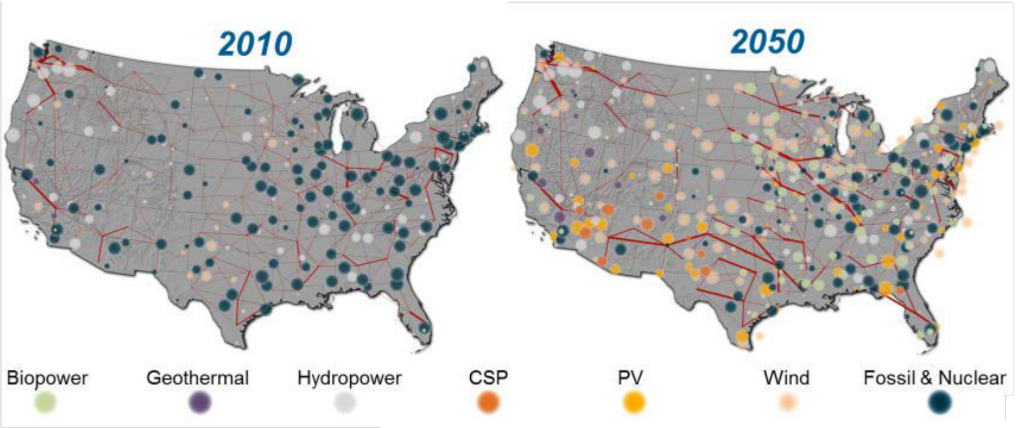
- **Audience:** Practitioners who would like to use ReEDS without a commercial linear programming solver. We assume the audience has some basic knowledge of:
  - (1) the ReEDS model (including the source code and executing the model),
  - (2) linear programming, and
  - (3) the GAMS software.

- **Key questions**:
  - What are some considerations for choosing a solver for ReEDS?
  - What are the practical steps to interface ReEDS with a solver?
  - Can an *open-source* solver compute solutions for the U.S. ReEDS model?

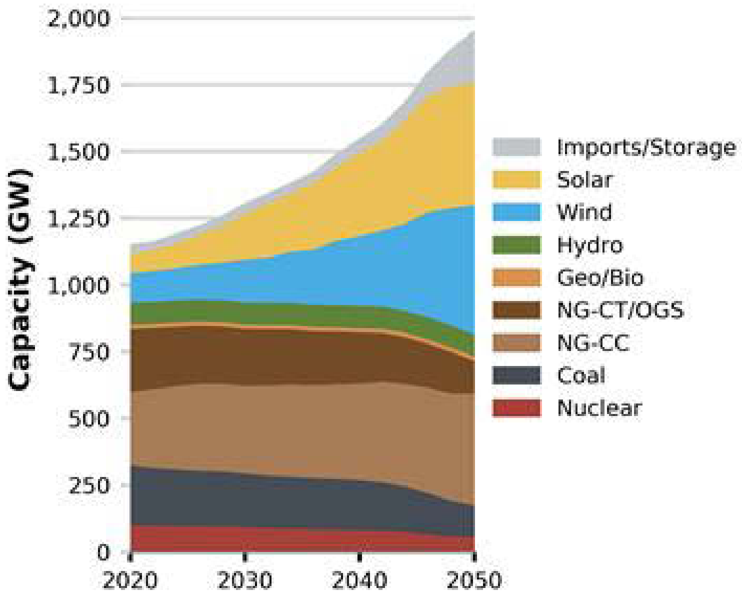
Regional Energy Deployment System (ReEDS) Model

- ReEDS is a publicly-available model developed at the National Renewable Energy Laboratory (NREL) that can be used to analyze the potential evolution of the U.S. electric power system. (https://www.nrel.gov/analysis/reeds)

- ReEDS is formulated as a linear program, written in the General Algebraic Modeling System (GAMS), and solved using a linear programming solver.



Source: NREL (2012).



Source: Cole et al. (2019).

Software license requirements for ReEDS

- While ReEDS is publicly available, the model requires a license for the GAMS Base Module (GBM) *and* a commercial linear programming solver.
  - The GBM includes features that are necessary to run ReEDS.
    - GAMS Language Complier and Execution System
    - GAMS Data Exchange utilities (e.g., GDXDUMP, GDXCOPY)
    - Links to free solvers (relevant for this tutorial)
  - NREL analysts use the commercial solver CPLEX via a GAMS/CPLEX license.

- Open-source solvers may serve as a viable alternative to commercial solvers like CPLEX.
  - **Pros**: Open-source solvers are typically free of charge to use.
  - **Cons**: Open-source solvers may have fewer features and/or less advanced algorithms compared to commercial solvers, and thus may have slower solve times.

Summary of key points

- We identify a list of linear programming solvers that (1) have low or no cost and (2) have established links with GAMS. We evaluate these solvers to identify potential alternatives to CPLEX for computing solutions for ReEDS.

- From this list, we select the COIN-OR Linear Programming (CLP) solver as the best candidate *open-source* software to use for ReEDS based on the solver's:
  - availability to all users at no cost,
  - inclusion of the interior point method,
  - ease of use within GAMS, and
  - performance in past solver benchmark studies.

Summary of key points (cont.)

- We benchmark the performance (solve time) of CPLEX and CLP for the reference, sequential-solve, U.S. ReEDS model version 2020.
  - As ReEDS progresses through the modeling horizon, the problem size grows. The larger problem size in later years leads to longer solve times compared with earlier years.
  - CPLEX finds solutions for each model year through 2050, with 20-130 seconds per solve.
  - CLP finds solutions for each model year through 2022, with 1,600-17,300 seconds per solve (0h27m – 4h48m).
  - In the 2024 model year, CLP fails to find an optimal solution within 30,000 seconds (8h20m), a time limit we impose for a single-year solve.

- The solution times for CLP *might* be improved by:
  - improving the numerical stability of ReEDS model instances, and/or
  - reducing the size of ReEDS model instances.

- Commercial solvers like CPLEX are especially useful if the problem size cannot be reduced and/or if solve time is an important factor.

Contents

Solver selection criteria

- We identify a list of linear programming solvers that (1) have low or no cost and (2) have established links with GAMS (see the table on the next slide).
  - Some of the solvers we identify are commercial software, but they include lower-cost academic and government licenses (relative to the cost of the commercial license).
  - Although some communities have access to these lower-cost options, our goal is to identify software that can be used by anyone.

- From this list, we select CLP as the best candidate for ReEDS based on the solver's:
  - availability to all users (free and open source),
  - inclusion of the interior point method (the recommended solution method for ReEDS),
  - ease of use within GAMS (a CLP solver link is available with the GBM), and
  - performance in past solver benchmark studies (see Appendix).

- The IPOPT solver and PIPS solver have similar merits to CLP, but:
  - preliminary test of IPOPT for ReEDS failed (see Appendix), and
  - the PIPS solver does not currently have a publicly available link with GAMS.

- *Note: CLP is the solver used to compute the linear programming relaxations for CBC (Coin-or Branch and Cut) a mixed integer linear programming solver https://projects.coin-or.org/Cbc.*

Solvers with (1) low or no cost and (2) established links with GAMS

Solver/Solver Link	Description	License	Webpage (Accessed 2021/06/03)	LP Algorithms	Notes
BDMLP	Brook, Drud, and Meeraus Linear Program Solver	GAMS Base Module License	https://new.gams.com/33/docs/S_BDMLP.html	Simplex	• BDMLP is not open source, but the solver is available with the GBM license. • As of GAMS version 34, BDMLP is *no longer* part of the GAMS distribution.
CLP	COIN-OR Linear Programming Solver	Eclipse Public License – v 2.0	https://github.com/coin-or/Cbc	1. Simplex 2. Interior Point	A Solver link is available with the GBM license.
GLPK	GNU Linear Programming Kit	GNU General Public License	https://www.gnu.org/software/glpk/	1. Simplex 2. Interior Point	GLPK is no longer part of the GAMS distribution.
IPOPT	Interior Point OPTmizer	Eclipse Public License	https://github.com/coin-or/Ipopt	Interior Point	A solver link is available with the GBM license.
MINOS (Academic or Government)	Modular In-core Nonlinear Optimization System	Custom license for Academic and Government	http://www.sbsi-sol-optimize.com/asp/sol_product_minos.htm	Two-phase primal Simplex	A GAMS/solver-link license is *not* available, but the solver can be access in GAMS with a GAMS/MINOS license.
PIPS	Parallel Interior Point Solver	Custom Open-Source License	https://github.com/Argonne-National-Laboratory/PIPS	1. Simplex 2. Interior Point	A PIPS-IPM solver link was developed for the BEAM-ME Project (2020) but is *not* publicly available.
OSI (Academic) – *Solver Links*	Open Solver Interface – bare-bones link to academic versions of commercial solvers (CPLEX, GUROBI, MOSEK, XPRESS)	Academic License	https://support.gams.com/solver:academic_programs_by_solver_partners https://www.gams.com/latest/docs/S_OSI.html	1. Simplex 2. Interior Point	The OSI is available through an GBM *Academic* License.
SOPLEX (Academic)	Sequential Object-oriented simPlex	ZIB Academic License	https://soplex.zib.de/	Simplex	A solver link is available with the GBM *Academic* License.

Contents

Using CLP for ReEDS

- A linkage between GAMS and open-source COIN-OR solvers like CLP is available with the GBM.

- Three steps are required to use CLP for ReEDS.
  1. Create a solver option file for CBC (“cbc.opt”).
  2. Specify “CBC” as the solver in the ReEDS scenario option files.
  3. Modify the GAMS infeasibility tolerance option.

(1) CBC solver option files

cbc.opt

```
startalg barrier
crash off
threads 8
```

Interior point method

No advanced basis

Multiple threads

\ReEDS_Augur\cbc.opt

```
startalg barrier
crash off
threads 8
```

- There are solver option files that exist in two locations:
  - ReEDS main file directory
  - Augur file directory
- We define a few basic CLP solver options, but users may customize these options (see Appendix).
- The interior point method (IPM) generally works well for the ReEDS problem structure.
  - When using the IPM, an advanced basis should *not* be used.
  - Parts of the IPM algorithm can be parallelized, so using multiple threads may be beneficial.

(2) ReEDS scenario option files

cases.csv

	Description	Default Value
timetype	Sequential or Intertemporal	seq
GAMSDIR	Directory for GAMS to be passed	c:\gams\gams\24.7
yearset_suffix	Set of years that are modeled	default
endyear	Last year to be modeled	2050
solver	Solver for GAMS to use	CBC

\ReEDS_Augur\values_defaults.csv

key	value	dtype	description
reedscc_szn_hours	10	int	number of top hours considered in seaso
remove_csv	TRUE	boolean	when true: delete Augur CSV files after it f
sleep_time	0.1	float	sleep time for skipping Augur steps - help
solver	CBC	str	solver for augur

- There are ReEDS scenario option files that exist in two locations:
  - ReEDS main file directory
  - Augur file directory
- Change the value for the "solver" row from "CPLEX" (default) to "CBC".
- The solver name provided in these files will be used in GAMS to specify the solver:
  - *d_solveoneyear.gms*
  - *B1_osprey.gms*

```
option LP = <solver> ;
```

(3) GAMS infeasibility tolerance

- We redefine the GAMS infeasibility tolerance to “1e-14” due to “empty row infeasibilities” that arise when using CLP.
- See the Appendix for an explanation of why these infeasibilities occur and why changing the GAMS infeasibility tolerance resolves the issue.
- If the infeasibilities persist, then the user should loosen the tolerance (i.e., make the value larger).

d_solveprep.gms

```
*==================================
* -- MODEL AND SOLVER OPTIONS --
*==================================


OPTION lp = %solver% ;
%case%.optfile = %GSw_gopt% ;
OPTION RESLIM = 50000 ;
*treat fixed variables as parameters
%case%.holdfixed = 1 ;


%case%.tolInfeas=1e-14;
```

Contents

Solver benchmark overview

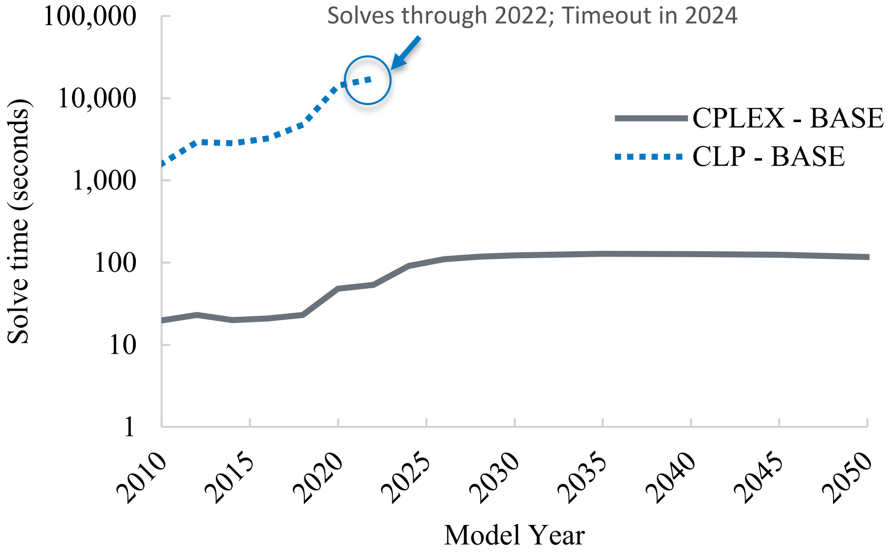
- We attempt to benchmark the CLP solver performance against CPLEX using the “reference sequential solve” scenario for ReEDS model version 2020.
  - See “ref_seq” in cases.csv.
  - https://github.com/NREL/ReEDS_OpenAccess/releases/tag/v2020.0
  - *Note: the github.com link will only work if you have requested access to the ReEDS repository.*

- We execute all test scenarios on a computer with:
  - Intel(R) Xeon(R) Gold 5120 CPU @ 2.20GHz, 14 Cores, 28 Logical Processors
  - 320 GB RAM

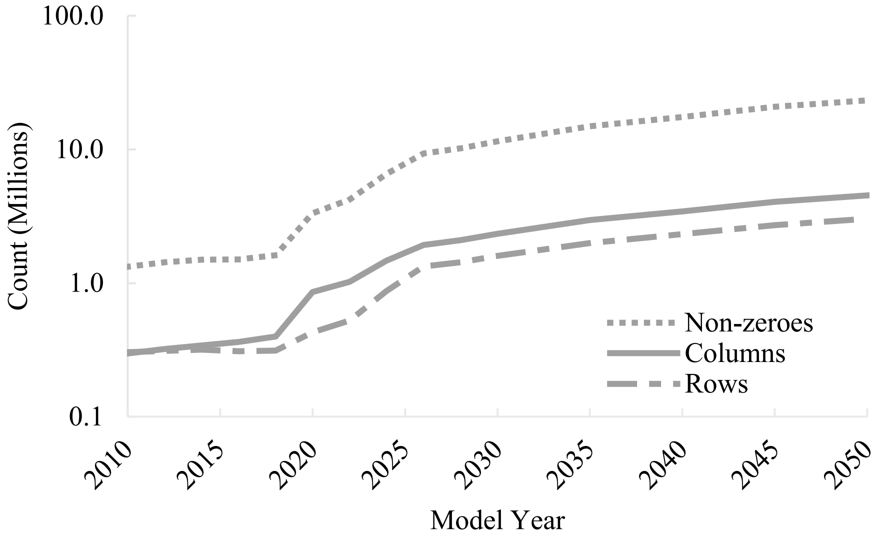
- We apply the same solver options for both CLP and CPLEX:
  - Interior point method
  - No advanced basis
  - Eight (8) threads

Solver benchmark results

Solve Times



Model size as reported by CPLEX (before CPLEX Presolve)



ReEDS 2050 Model instance	rows	columns	non-zeros
US – before CPLEX Presolve	3.0M	4.5M	23.3M
US – after CPLEX Presolve	0.44M	0.52M	2.4M

Solver benchmark results

- As a ReEDS sequential-solve case progresses through the modeling horizon, the A-matrix size increases (rows; columns; non-zeros), yielding longer solve times in later years versus earlier years.
  - New investments are limited to exogenous capacity prescriptions in historical model years (2010-2018).
  - The number of variables for tracking capital stock increases beginning in the 2020 decade.
    - Endogenous investments are enabled throughout the 2020 decade (the year is specific to each technology).
    - Endogenous retirements are enabled in 2024.
  - The number of model plants increases as the model progresses into future years.
    - Model plants are tracked based on when they are built (vintage).

Solver benchmark results (cont.)

- CPLEX finds solutions for each model year through 2050, with 20-130 seconds per solve.
- CLP finds solutions for each model year through 2022, with 1,600-17,300 seconds per solve (0h27m – 4h48m).
- In 2024, CLP exceeds the 30,000 second (8h20m) time limit that we impose for each single-year solve. We increase the limit to 100,000 seconds (27h47m), but CLP exceeds this as well.

- There are ways to potentially improve the solve times, which we explore in the next section.

Contents

Improving solve times

- **Relevant Literature**
  - *Klotz and Newman (2013)* offer practical guidelines for improving solve times for linear programs; some guidelines are specific to CPLEX, but the authors also propose solver-agnostic best practices.
  - *Scholz et al. (2020)* summarize best practices for speeding up solve times for energy system models, including reducing the model complexity and exploiting the problem structure using the PIPS-IPM solver linked to GAMS within a high-performance computing environment.

- **Approach 1: Improve the A-matrix.**
  - Adjust the matrix coefficients to improve numerical stability.
    - Round very small coefficients to zero.
    - Scale the coefficients.
  - Reformulate the problem

- **Approach 2: Reduce the size of the A-matrix.**
  - Remove variables and constraints by turning “off” model features.
  - Reduce the model dimensions.

Improve the A-matrix

1. Round the A-matrix coefficients
2. Scale the A-matrix coefficients
3. Reformulate the problem

(1) Round the A-matrix coefficients

- **Example**:
  - A 100 MW PV system that produces an average of 0.003 MW within a time-slice has a capacity factor of 0.00005.
  - In this case, we can round the capacity factor to zero and assume the PV system does not produce any energy during this time-slice.

- Users must determine when rounding is appropriate and how many decimal points to round.
  - The number of decimal points should be unique to each parameter.
  - Incorrect rounding may lead to inaccurate solutions (see Klotz and Newman 2013).

(2) Scale the A-matrix coefficients

- **Issue**: Certain instances of ReEDS can be difficult to solve due to poorly scaled A-matrix coefficients.
- **Goal**: Improve the A-Matrix coefficients by moving them as close to unity as possible.
- **Best Practice**: Limit 12 orders of magnitude between the absolute value of the smallest and largest coefficients.
  - Ideally, this spread should be as small as possible.
  - See Klotz and Newman (2013) for more information.
- **Approach**: Inspect the A-Matrix to identify poorly scaled coefficients and make scaling adjustments. See the Appendix for some tools for doing this inspection.
- **Note**: Scaling the coefficients in one part of the matrix can affect scaling in another part.

Examples for how to scale the A-matrix coefficients

1. Automatic scaling using solver options (see Appendix).
2. Manual scaling using the GAMS ".scale" feature.
   - `<constraint>.scale`
   - `<variable>.scale`
3. Manual scaling with user-defined scaling parameters
   - **ReEDS Example**: apply a scaling parameter for emissions variables that are specific to each pollutant: `emit_scale(e)`
   - `EMIT(i,e)=emit_scale(e)*emit_rate(i,e)*GEN(i)`

(3) Reformulate the problem

- Literature shows that reformulating a linear program can be an effective tool for improving the solvability of a problem instance.
  - Trick (2005), Bertsimas et al. (2010), Klotz and Newman (2013), Newman and Weiss (2013), Lambert et al. (2014)

- **ReEDS Example**: Transmission flow limits are imposed in either direction along a ReEDS transmission corridor. Due to transmission losses, we do not expect bi-directional flows. Therefore, we can propose a simple reformulation. This reformulation assumes that the transmission capacity is identical in either direction.

Old Constraints :

```
FLOW(r,r’) <= CAP(r,r’)
FLOW(r’,r) <= CAP(r’,r)
```

New Constraints:

```
FLOW(r,r’) + FLOW(r’,r) <= CAP(r,r’) such that ord(r) > ord(r’)
```

Reduce the size of the A-matrix

1. Remove model features
2. Reduce the model dimensions

(1) Remove model features

- For certain ReEDS scenarios, some constraints may not be binding.
  - Therefore, removing these constraints *a priori* can reduce the problem size.
  - Although some constraints may not be very impactful in one ReEDS scenario, these constraints may be impactful in other scenarios.

- By removing ReEDS model features, the practitioner is making a compromise on model detail with the intent of improving the solve time.

- Some ReEDS model features can be turned “on” or “off” using the “switches” in the “cases.csv” file.

  https://github.com/NREL/ReEDS_OpenAccess#Switches

  Note: github.com link will only work if you have requested access to the ReEDS repository.

(1) Remove model features

Category	ReEDS Model Feature	ReEDS Switch (cases.csv)
Emissions Policy (EMIS)	AB32 CO2 Limit	GSw_AB32
	RGGI CO2 Limit	GSw_RGGI
	CSAPR SOx and NOx Limits	GSw_CSAPR
Capital Stock (CAP)	Endogenous Refurbishments	GSw_Refurb
	Endogenous Retirements	GSw_Retire
CCS Technology (CCS)	Carbon Capture and Storage	GSw_CCS
RPS Policy (RPS)	State Renewable Portfolio Standards	GSw_StateRPS
Operating Reserves (OR)	Operating Reserves	GSw_Opres

- These are a subset of model features in “cases.csv” that are “on” by default.
- In the next section, we demonstrate the impact on the solve time that results from turning these features “off”.

(2) Reduce the model dimensions

- Examples of ReEDS model dimensions include (but are not limited to) *regions* and *years*.
- By reducing the dimensionality of ReEDS, the practitioner is making a compromise on model detail with the intent of improving the solve time.

ReEDS Model Dimension	Default Configuration	Dimension Reduction	Compromise
Regions	The lower 48 United States has 134 balancing areas	ERCOT has 7 balancing areas	Only model a single interconnect.
Years	2-year solves from 2010-2030 5-year solves from 2035-2050	10-year solves from 2010-2050	Do not capture time-varying input parameters, e.g., changes in federal tax credits.

Test Scenarios

Test Scenarios

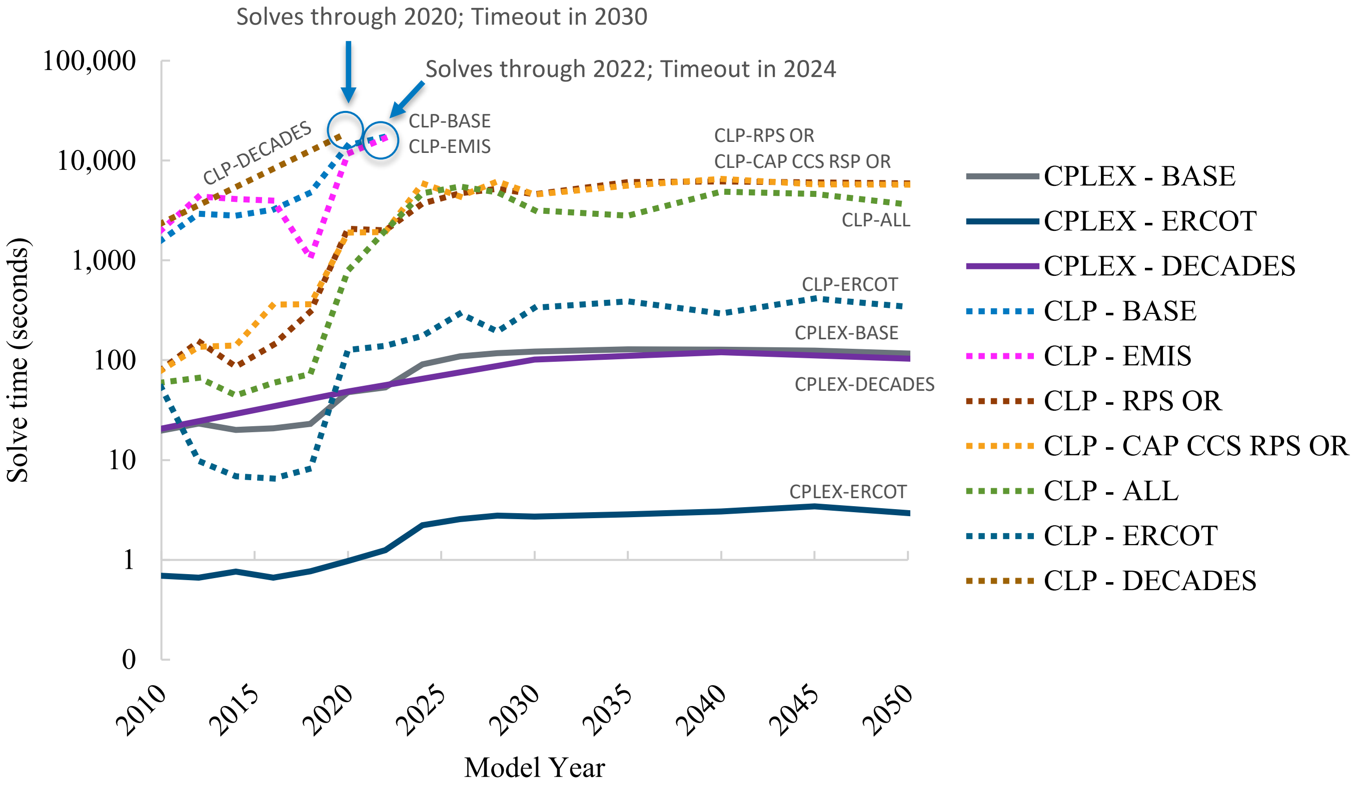
- We test some of the techniques described in the previous sections.
- Evaluation metrics include impact on the solve time and problem size (rows, columns, non-zeros).

Category	Technique	ReEDS Example	Do we test this?
(1) Improve the A-matrix	Round coefficients	Round the emission rate parameters	YES
	Scale coefficients	Scale the emissions variables	YES
	Reformulate the problem	Reformulate the constraints for transmission limits	No
(2) Reduce the size of the A-matrix	Remove model features	Turn “off” endogenous retirements	YES
	Reduce the model dimensions	• Use the ERCOT model. • Make investment decisions once per decade	YES

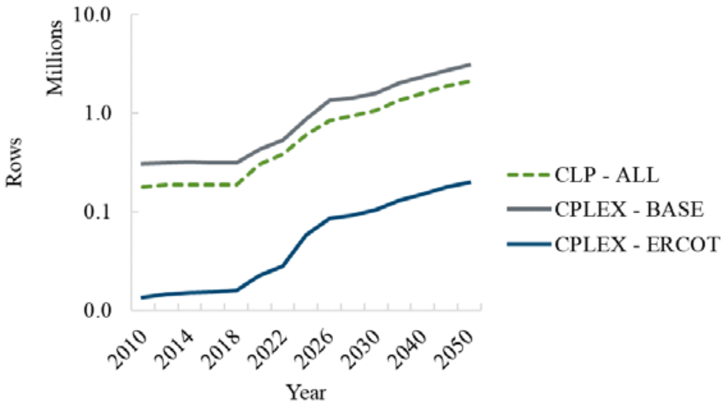
Definitions of the test scenarios

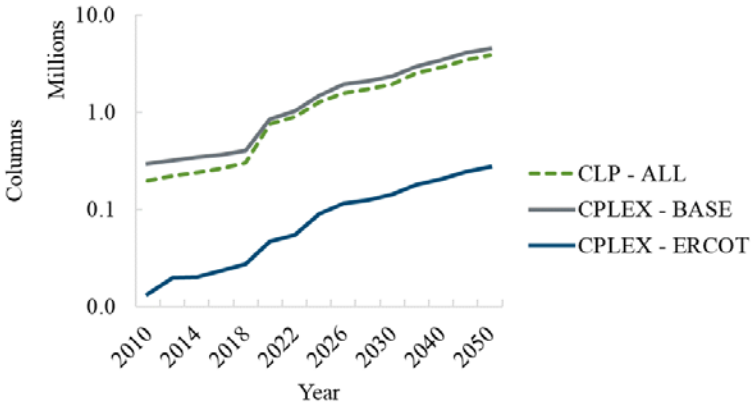
Scenario Name	Category and Technique	Description	Impact on solve time relative to BASE
BASE	N/A	Default U.S. ReEDS model Version 2020 (“ref_seq”)	Timeout in 2024
Emit_rate	1. Improve the A-matrix: *rounding*	Set the rounding of the emission rate parameter (emit_rate) to a few decimal points	• No solve time improvement through 2022 • Timeout in 2024
Emit_scale	1. Improve the A-matrix: *scaling*	Scale the emissions variables and constraints using a scaling parameter that is a specific to the pollutant type	• No solve time improvement through 2022 • Timeout in 2024
EMIS	2. Reduce the A-matrix: *model features*	Turn “off” constraints for emissions	• No solve time improvement through 2022 • Timeout in 2024
RPS OR	2. Reduce the A-matrix: *model features*	Turn “off” constraints for state RPS and operating reserves	• 10x reduction in solve time through 2022 • Solves successfully to 2050
CAP CCS RPS OR	2. Reduce the A-matrix: *model features*	Turn “off” constraints for capital stock, CCS technologies, state RPS, and operating reserves	• 10x reduction in solve time trough 2022 • Solves to 2050
ALL (EMIS CAP CCS RPS OR)	2. Reduce the A-matrix: *model features*	Turn “off” all model features listed on Slide 29	• 10x reduction in solve time through 2022 • Solves to 2050
ERCOT	2. Reduce the A-matrix: *model dimensions*	Reduce the spatial extent to the ERCOT system	• Significant reduction in solve time from BASE • Solves to 2050
DECADES	2. Reduce the A-matrix: *model dimensions*	Solve the model every ten years	• Similar solve times in 2010 and 2020 as BASE • Timeout in 2030

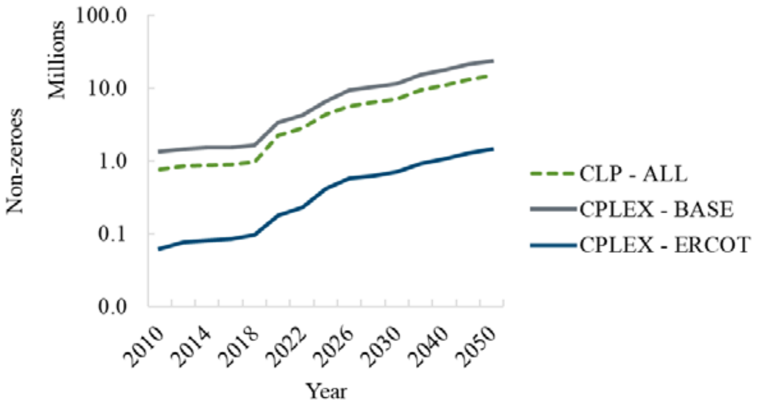
Solution times



Problems sizes







Contents

Recommendations

- Consider using CLP for ReEDS if your analysis permits simplifications.
- Although turning “off” RPS constraints and operating reserve constraints is effective in reducing the CLP solve time, these two model features may be critical for producing useful analysis.
  - Excluding RPS constraints may lead to non-compliance with state-specific renewable generation requirements.
  - Excluding operating reserve constraints may result in a capacity mix that is unable to meet expected or unexpected changes in generation and load.
  - Consider turning “off” other model features not included in this tutorial.
  - Consider trying combinations of turning “off” model features *and* reducing model dimensions.
- Consider using a commercial solver if the problem size cannot be reduced and/or if solve time is an important factor.

Possible Future Work

- Test other open-source solvers beyond CLP, especially IPOPT. Additional work is required to make IPOPT compatible with ReEDS.
- Execute CLP with an increased number of threads.
- Expand the number of model features that are controlled by "switches" to make it easier for users to make simplifications to ReEDS scenarios.
- Continue to improve the numerical stability of ReEDS through scaling. Additionally, identify opportunities for reformulation.
- Use a commercial solver to execute a broad range of simplified ReEDS scenarios and analyze the impacts of the simplifications on the model solution (both in isolation and in combination).

References

D. Bertsimas, D. B. Brown, and C. Caramanis, “Theory and Applications of Robust Optimization,” *ArXiv10105445 Cs Math*, Oct. 2010, Accessed: Apr. 23, 2021. [Online]. Available: http://arxiv.org/abs/1010.5445.

The BEAM-ME Project (2020). “Implementation of acceleration strategies from mathematics and computational sciences for optimizing energy system models”. http://www.beam-me-projekt.de/beam-me/EN/Home/home_node.html. Accessed Septembe 26, 2020.

Bussieck and Vigerske (2007). *Interfacing COIN-OR solvers by GAMS.* https://www.gams.com/archives/presentations/present_gamslinks.pdf. Accessed September 26, 2020.

COIN-OR (2011). *Benchmarks of GAMS solvers.* https://www.coin-or.org/GAMSlinks/benchmarks/. Accessed September 26, 2020.

Cole, Wesley J., Nathaniel Gates, Trieu T. Mai, Daniel Greer, and Paritosh Das. (2019). 2019 Standard Scenarios Report: A US Electric Sector Outlook. No. NREL/PR-6A20-74110. National Renewable Energy Laboratory, Golden, CO (United States), 2020.

Gearhart, Jared L., Kristin L. Adair, Richard J. Detry, Justin D. Durfee, Katherine A. Jones, and Nathaniel Martin. "Comparison of open-source linear programming solvers." Sandia National Laboratories, SAND2013-8847 (2013). https://prod-ng.sandia.gov/techlib-noauth/access-control.cgi/2013/138847.pdf

Gillig, Dhazn and McCarl, Bruce A. (n.d.) *An introduction to GAMSCHK.* https://agecon2.tamu.edu/people/faculty/mccarl-bruce/641clas/08_641_gams_check.pdf

Klotz, Ed, and Alexandra M. Newman. (2013). "Practical guidelines for solving difficult mixed integer linear programs." Surveys in Operations Research and Management Science 18, no. 1-2 (2013): 18-32.

W. B. Lambert, A. Brickey, A. M. Newman, and K. Eurek, “Open-Pit Block-Sequencing Formulations: A Tutorial,” *Inf. J. Appl. Anal.*, vol. 44, no. 2, pp. 127–142, Apr. 2014, doi: 10.1287/inte.2013.0731.

Mittelmann, Hans. (2020). “Decision Tree for Optimization Software: Benchmarks for Optimization Software”. Accessed September 26, 2020. http://plato.asu.edu/bench.html.

National Renewable Energy Laboratory. (2012). Renewable Electricity Futures Study. Hand, M.M.; Baldwin, S.; DeMeo, E.; Reilly, J.M.; Mai, T.; Arent, D.; Porro, G.; Meshek, M.; Sandor, D. eds. 4 vols. NREL/TP-6A20-52409. Golden, CO: National Renewable Energy Laboratory. http://www.nrel.gov/analysis/re_futures/.

A. M. Newman and M. Weiss, “A Survey of Linear and Mixed-Integer Optimization Tutorials,” *Inf. Trans. Educ.*, vol. 14, no. 1, pp. 26–38, Sep. 2013, doi: 10.1287/ited.2013.0115.

Scholz, Yvonne, Benjamin Fuchs, Frieder Borggrefe, Karl-Kien Cao, Manuel Wetzel, Kai von Krbek, Felix Cebulla et al. "Speeding up Energy System Models-a Best Practice Guide." (2020).

M. Trick, “Formulations and Reformulations in Integer Programming,” in *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, Berlin, Heidelberg, 2005, pp. 366–379, doi: 10.1007/11493853_27

Contact: kelly.eurek@nrel.gov

www.nrel.gov

NREL/PR-6A20-77907

This work was authored by the National Renewable Energy Laboratory, operated by Alliance for Sustainable Energy, LLC, for the U.S. Department of Energy (DOE) under Contract No. DE-AC36-08GO28308. Funding provided by the U.S. Department of Energy Office of Strategic Programs. The views expressed in the article do not necessarily represent the views of the DOE or the U.S. Government. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paid-up, irrevocable, worldwide license to publish or reproduce the published form of this work, or allow others to do so, for U.S. Government purposes.

NREL Transforming ENERGY

Appendix: Solver Benchmark Studies

Benchmark studies for LP solvers

- Bussieck and Vigerske (2007). *Interfacing COIN-OR solvers by GAMS.*
  - https://www.gams.com/archives/presentations/present_gamslinks.pdf
  - CLP had the highest percentage of models solved among other open-source solvers (GLPK and IPOPT).

- COIN-OR (2011). *Benchmarks of GAMS solvers.*
  - https://www.coin-or.org/GAMSlinks/benchmarks/
  - Solvers tested: CPLEX, CLP, GLPK, GUROBI, MOSEK
  - CLP performed third best behind CPLEX and GUROBI for problem instances above 100k constraints, 100k variables, and 1M non-zeros.
  - CLP solve times were about double those of CPLEX and GUROBI.
  - https://www.coin-or.org/GAMSlinks/benchmarks/LP/allSolver_110307/timings.htm

- Gearhart et al. (2013). *Comparison of open-source linear programming solvers.*
  - CLP performed the best in "capability and speed".
  - GLPK, LP_SOLVE, MINOS (commercial) were also considered.

- Mittelmann (2020). *Decision Tree for Optimization Software: Benchmarks for Optimization Software.*
  - http://plato.asu.edu/bench.html
  - Interior point solvers tested: MOSEK, MATLAB (no xover), CLP, SAS-OR, Tulip (no xover); http://plato.asu.edu/ftp/lpbar.html
  - Generally, CLP finished last among the solvers for problem instances above 100k constraints, 100k variables, and 1M non-zeros, although not all solvers used crossover which can be time intensive.
  - CLP solved a large problem instance (1.15M constraints, 747.7k variables, and 4.7M non-zeros) where two other solvers failed.
  - CLP failed to solve the largest problem instance (3M constraints, 1.43M variables, 14.2M non-zeros).

Appendix: Initial Test with IPOPT

Initial Tests with IPOPT

- IPOPT will solve the ERCOT 2010 model instance.

- However, then ReEDS fails in the Augur Module due to a problem with the ReEDS data that is sent to Augur. Specifically, IPOPT assigns levels to variables that are excluded from the optimization.

- Example: ReEDS ERCOT 2010 model instance:
  - IPOPT finds a solution with 10 GW of new CSP investments in 2010, but new CSP investments are not permitted in historical years.
  - The investment variable levels for CSP are equal to 10,000
    - INV("csp-ns","new-1","s20","2010")=10000
  - However, these variables do not show up in the objective function nor the A-matrix because they are not valid investment options in 2010.
    - valinv("csp-ns","new-1","s20","2010")=0.

```
*investment costs
                  sum{(i,v,r)$valinv(i,v,r,t),
                    INV(i,v,r,t) * (cost_cap_fin_mult(i,r,t) * cost_cap(i,t)
```

Appendix: Solver Options

Comparison of solver options for CPLEX and CLP

Option	CPLEX https://www.gams.com/latest/docs/S_CPLEX.html	CLP https://www.gams.com/latest/docs/S_CBC.html
Use barrier method for LP	Lpmethod 4	Startalg barrier
Do crossover	Solution Type 1	Crossover 1
Do not crash the basis	Avdind 0	Crash 0
Specifying multiple threads	Threads	Threads
Scale the matrix	Scaind 0 or Scaind 1	Scaling auto
For an optimal solution, no dual infeasibility may exceed this value	eprhs	Tol_dual 1e-7
For a feasible solution, no primal infeasibility may exceed this value		Tol_primal 1e-7
Tolerance to use in presolve		Tol_presolve 1e-8
Write MPS file (for diagnostics)	writemps writepre	writemps

Empty Row Infeasibility

Empty Row Infeasibility

- In the 2016 model year, we see an infeasibility error.
- Infeasible constraints: “eq_rsc_INVlim”

Investments in each bin <= bin capacity

```
**** Exec Error at line 1659962: Equation infeasible due to rhs value

**** INFEASIBLE EQUATIONS ...
GAMS 30.3.0  rc5da09e Released Mar  6, 2020 WEX-WEI x86 64bit/MS Windows
'ReEDS 2.0'
Model Analysis      SOLVE cbc_jul19_ref_seq Using LP From line 1662529


---- eq_rsc_INVlim  =G=  --MW-- total investment from each rsc bin cannot exceed the available investment

eq_rsc_INVlim(p11,upv_7,bin2,2016)..  0 =G= 7.105427357601E-15 ; (LHS = 0, INFES = 7.105427357601E-15 ****)

eq_rsc_INVlim(p18,hydUND,bin2,2016)..  0 =G= 1.77635683940025E-15 ; (LHS = 0, INFES = 1.77635683940025E-15 ****)

eq_rsc_INVlim(p110,hydNPND,bin2,2016)..  0 =G= 1.77635683940025E-15 ; (LHS = 0, INFES = 1.77635683940025E-15 ****)
```

Empty Row Infeasibility

- **Example**: `eq_rsc_INVlim(UPV_7, p11, bin2, 2016)`

- Resource supply curve bin capacities
  - Bin 1: 6.9984 MW
  - Bin 2: 38.1024 MW
  - Bin 3: 14.5152 MW

- There are three historical capacity prescriptions through 2014
  - Total = 46.8 + 8.19 = 54.99 MW
  - The prescriptions use all capacity from supply curve bins 1 and 2 and use most capacity from bin 3.

Empty Row Infeasibility

- By default, we use the GAMS "holdfixed" option to convert fixed variables to constants.

- **Infeasibility for eq_rsc_INVlim(UPV_7, p11, bin2, 2016)**

  0 >= m_rsc_dat(bin2,"cap") – $\sum_{t' \leq 2016}$ INV.fx(bin2, t')

  Where,

  $\sum_{t' \leq 2016}$ INV.fx(bin2, t') = 38.102400000000007
  m_rsc_data(bin2,"cap") = 38.1024

  Thus, the constraint evaluates to: 0 >= 7e-15

- The RHS evaluates to a very small number (7e-15) which makes the constraint infeasible according to the GAMS infeasibility tolerance.

Empty Row Infeasibility

- This infeasibility is unique because we have an "empty row".
- The default GAMS feasibility tolerance is 10 times the machine precision.
- Proposed solution:
  - Loosen the feasibility tolerance in GAMS.
  - `<model>.tolinfeas = <#> ;`
  - https://www.gams.com/32/docs/UG_GamsCall.html#GAMSAOtolinfeas

GAMS Feasibility Example:

```
variable z ;
equation obj, error ;

obj.. z=e=1;
error.. 0 =g= 7e-15 ;

model test /all/ ;

* FEASIBLE
test.tolinfeas = 1e-14 ;

* INFEASIBLE
*test.tolinfeas = 1e-15 ;

solve test using lp min z ;
```

Inspecting the A-Matrix

Mathematical Programming System (MPS)

- MPS files can be used to explore A-matrix, b-vector, and c-vector coefficients.
- MPS is a standardized format used to store an LP problem instance, including, A-matrix, b-vector, and c-vector coefficients, and variable bounds.

Sample List of Constraints

```
ROWS
 N   OBJROW
 E   eq_loadcon(p1,h1,2014)
```

Sample list of Variables, A-matrix coefficients, and c-vector coefficients

```
COLUMNS
    LOAD(p1,h1,2014)  eq_loadcon(p1,h1,2014)  1.             eq_supply_demand_balance(p1,h1,2014)   -1.
    LOAD(p1,h1,2014)  eq_OpRes_requirement(reg,p1,h1,2014)  -0.01         eq_OpRes_requirement(spin,p1,h1,2014)  -0.03
```

Sample list of b-vector coefficients

```
RHS
    RHS  eq_loadcon(p1,h1,2014)  4982.665486   eq_loadcon(p1,h2,2014)  6098.3130782
```

Mathematical Programming System (MPS)

- MPS files can be used to explore A-matrix, b-vector, and c-vector coefficients.
- MPS is a standardized format used to store an LP problem instance, including, A-matrix, b-vector, and c-vector coefficients, and variable bounds.

Variable

Constraint where the variable appears

A-matrix or c-vector coefficient

Sample List of Constraints

```
ROWS
 N   OBJROW
 E   eq_loadcon(p1,h1,2014)
```

Sample list of Variables, A-matrix coefficients, and c-vector coefficients

```
COLUMNS
    LOAD(p1,h1,2014)  eq_loadcon(p1,h1,2014)  1.             eq_supply_demand_balance(p1,h1,2014)   -1.
    LOAD(p1,h1,2014)  eq_OpRes_requirement(reg,p1,h1,2014)  -0.01        eq_OpRes_requirement(spin,p1,h1,2014)  -0.03
```

Sample list of b-vector coefficients

```
RHS
    RHS eq_loadcon(p1,h1,2014)  4982.665486   eq_loadcon(p1,h2,2014)  6098.3130782
```

B-vector coefficient

Constraint

Inspect the A-matrix: (1) Examine an MPS files using the GAMS IDE.

- Open the MPS file in the GAMS Interactive Development Environment (IDE).
- Search for large (e7, e8, …) and small exponents (e-7, e-8, …).
- The image below is a snapshot of an MPS file for ReEDS with very small coefficients for some of the emission-related constraints.

```
GEN(Gas-CC,init-1,p1,h15,2014)  eq_capacity_limit(Gas-CC,init-1,p1,h15,2014)  -1.          eq_emit_accounting(CO2,p1,2014)  -0.000146133
GEN(Gas-CC,init-1,p1,h15,2014)  eq_emit_accounting(SO2,p1,2014)      -4.416e-9  eq_emit_accounting(NOX,p1,2014)     -2.76e-8
GEN(Gas-CC,init-1,p1,h15,2014)  eq_gasaccounting_regional(PA,2014)   -2752.64    eq_gasaccounting_national(2014)    -2752.64
GEN(Gas-CC,init-1,p1,h16,2014)  OBJROW 215802.25289 eq_supply_demand_balance(p1,h16,2014) 1.
GEN(Gas-CC,init-1,p1,h16,2014)  eq_capacity_limit(Gas-CC,init-1,p1,h16,2014)  -1.          eq_emit_accounting(CO2,p1,2014)  -0.000182666
GEN(Gas-CC,init-1,p1,h16,2014)  eq_emit_accounting(SO2,p1,2014)       -5.52e-9  eq_emit_accounting(NOX,p1,2014)     -3.45e-8
GEN(Gas-CC,init-1,p1,h16,2014)  eq_gasaccounting_regional(PA,2014)    -3440.8    eq_gasaccounting_national(2014)    -3440.8
GEN(Gas-CC,init-1,p1,h17,2014)  OBJROW 18765.413295 eq_supply_demand_balance(p1,h17,2014) 1.
GEN(Gas-CC,init-1,p1,h17,2014)  eq_capacity_limit(Gas-CC,init-1,p1,h17,2014)  -1.          eq_emit_accounting(CO2,p1,2014)    -1.5884e-5
GEN(Gas-CC,init-1,p1,h17,2014)  eq_emit_accounting(SO2,p1,2014)       -4.8e-10  eq_emit_accounting(NOX,p1,2014)        -3e-9
GEN(Gas-CC,init-1,p1,h17,2014)  eq_gasaccounting_regional(PA,2014)    -299.2     eq_gasaccounting_national(2014)    -299.2
GEN(Gas-CC,init-1,p2,h1,2014)  OBJROW 342865.25572 eq_supply_demand_balance(p2,h1,2014) 1.
GEN(Gas-CC,init-1,p2,h1,2014)  eq_capacity_limit(Gas-CC,init-1,p2,h1,2014)  -1.          eq_emit_accounting(CO2,p2,2014)  -0.000291603
GEN(Gas-CC,init-1,p2,h1,2014)  eq_emit_accounting(SO2,p2,2014)      -8.832e-9  eq_emit_accounting(NOX,p2,2014)     -5.52e-8
GEN(Gas-CC,init-1,p2,h1,2014)  eq_gasaccounting_regional(PA,2014)   -5497.92    eq_gasaccounting_national(2014)    -5497.92
GEN(Gas-CC,init-1,p2,h2,2014)  OBJROW 300007.09876 eq_supply_demand_balance(p2,h2,2014) 1.
GEN(Gas-CC,init-1,p2,h2,2014)  eq_capacity_limit(Gas-CC,init-1,p2,h2,2014)  -1.          eq_emit_accounting(CO2,p2,2014)  -0.000255153
GEN(Gas-CC,init-1,p2,h2,2014)  eq_emit_accounting(SO2,p2,2014)      -7.728e-9  eq_emit_accounting(NOX,p2,2014)     -4.83e-8
```

Inspect the A-Matrix: (2) Examine an MPS file using python.

- Parse MPS files using functions from NREL's *Raw_Value_Streams.py* script (available in ReEDS 2.0 repository).

- Filter coefficients that are above and below user-specified thresholds.

- Report the filtered results for inspection.

```
import raw_value_streams as rvs
import pandas as pd

mpsfile=rvs.get_df_mps('mpsfile.mps')
```

```
def high_values(df,max_val):
    high_values=df[df['coeff'].abs()>max_val].copy()
    return high_values

def low_values(df,min_val):
    low_values=df[abs(df['coeff'])<min_val].copy()
    return low_values

mpsfile_high=high_values(mpsfile,1000000)
mpsfile_low=high_values(mpsfile,.000001)
```

```
pd.DataFrame.to_csv(mpsfile_high,'mpsfile_high.csv')
pd.DataFrame.to_csv(mpsfile_low,'mpsfile_low.csv')
```

Inspect the A-Matrix: (3) Produce BLOCKPIC outputs using GAMSCHK.

- GAMSCHK is a GAMS utility available through the GBM.
- The "D. Scaling" output gives the minimum and maximum coefficients for each (row, column) pair. The figure on the right is an example for the classic transportation problem (trnsport.gms). https://www.gams.com/latest/gamslib_ml/libhtml/gamslib_trnsport.html
- This approach may not be practical to visualize for ReEDS given the large number of constraint and variable "blocks".

Blockpic

D. Scaling - Maximum & Minimum Coefficients by Block -- Strip 1

		Transport	TotalCost	RHS MaxMin	Equ MaxMin
CostSum	Max	250	1		250
	Min	151	1		1
SupplyBal	Max	1		600	1
	Min	1		350	1
DemandBal	Max	1		325	1
	Min	1		275	1
Total Var	Max	250	1	600	
	Min	1	1	275	

Source: Gillig, Dhazn and McCarl, Bruce, A. (n.d.)