Title

An open-source Python implementation of California's hydroeconomic optimization model

Permalink

https://escholarship.org/uc/item/430633nd

Authors

Dogan, Mustafa S

Fefer, Max A

Herman, Jonathan D

et al.

Publication Date

2018-10-01

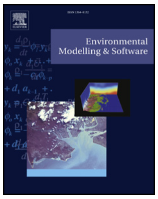
DOI

10.1016/j.envsoft.2018.07.002

Contents lists available at ScienceDirect

Environmental Modelling & Software

journal homepage: www.elsevier.com/locate/envsoft



An open-source Python implementation of California's hydroeconomic optimization model

Mustafa S. Dogan[a,b], Max A. Fefer[a,b], Jonathan D. Herman[a,b,*], Quinn J. Hart[c], Justin R. Merz[c], Josue Medellín-Azuara[d], Jay R. Lund[a,b]

[a] Department of Civil & Environmental Engineering, University of California, Davis, USA
[b] Center for Watershed Sciences, University of California, Davis, USA
[c] Department of Land, Air, and Water Resources, University of California, Davis, USA
[d] School of Engineering, University of California, Merced, USA

ARTICLE INFO

Keywords:
Hydroeconomic modeling
Network flow optimization
Multi-reservoir operations

ABSTRACT

This short communication describes a new open-source implementation of the CALVIN model (CALifornia Value Integrated Network), a large-scale network flow optimization model of California's water supply system. The model is cross-platform, uses common data formats, and connects to several freely available linear programming solvers. Given inputs including hydrology, urban/agricultural demand curves, and variable operating costs, the model minimizes the systemwide cost of water scarcity and operations including surface and groundwater reservoirs, wastewater reuse, desalination, environmental flow requirements, and hydropower. Key outputs include water shortage costs and marginal economic values of water and infrastructure capacity. We benchmark the scalability of different solvers up to roughly 5 million decision variables, using shared-memory parallelization on a high performance computing cluster. Runtimes are reduced by two orders of magnitude relative to the original model when no initial solution is provided, in addition to the benefits such as accessibility and transparency that come with an open-source platform. While this model is specific to California, the data and model structure are separated, so a similar framework could be used in any system where water allocation has been formulated as a network flow problem.

1. Background

1.1. Hydroeconomic modeling

Hydroeconomic models combine water resources systems engineering and economics, where water allocations are driven by economic value, and conversely, economic costs and benefits are impacted by hydrology (Cai, 2008; Harou et al., 2009; Booker et al., 2012). Optimization models incorporating economic objectives have been used extensively to evaluate water resources planning and management decisions over the past decades (Labadie, 2004). Draper et al. (2003) first introduced CALVIN (CALifornia Value Integrated Network), a hydroeconomic model describing California's water supply infrastructure including surface and groundwater reservoirs, urban and agricultural demands, environmental flow requirements, hydropower production, wastewater reuse and desalination facilities, plus urban and agricultural water conservation. The model has since been used in numerous studies, with topics ranging from climate adaptation to groundwater overdraft. Subsequent authors have updated the structure and parameters of the model to reflect changes in water demands and environmental requirements. The economic optimization framework is unique among statewide California models and provides results such as willingness-to-pay for additional water delivery and marginal value of increased conveyance and storage capacities. Other large-scale water resources studies in California have included contributions to groundwater modeling (Dogrul et al., 2016), agricultural economics (Howitt et al., 2012; Winter et al., 2017), and multi-objective analysis (Quinn et al., 2004; Yang et al., 2015).

The vast improvements in computing power since CALVIN's inception and its ability to explore potential scenarios in California water supply provide an opportunity to move the network structure and data to a new optimization platform. This short communication describes the outcome of this effort, following some general design goals:

[*] Corresponding author. Department of Civil & Environmental Engineering, University of California, Davis, USA.

E-mail addresses: msdogan@ucdavis.edu (M.S. Dogan), mafefer@ucdavis.edu (M.A. Fefer), jdherman@ucdavis.edu (J.D. Herman), qjhart@ucdavis.edu (Q.J. Hart), jrmerz@ucdavis.edu (J.R. Merz), jmedellin-azuara@ucmerced.edu (J. Medellín-Azuara), jrlund@ucdavis.edu (J.R. Lund).

https://doi.org/10.1016/j.envsoft.2018.07.002
Received 17 February 2018; Received in revised form 12 June 2018; Accepted 5 July 2018
Available online 12 July 2018

Software availability

Name of Software CALVIN (Python version)

Description An open-source hydroeconomic model that optimizes water allocation to agricultural and urban users in California, originally developed by Draper et al. (2003). Built on top of the Pyomo library (Hart et al., 2012). Storage and demand nodes are connected in a network structure; the dataset is hosted here: (https://github.com/ucd-cws/calvin-network-data)

Developers M. Dogan, M. Fefer, J. Herman, Q. Hart, J. Merz

Funding Source Development was partially supported by the California Energy Commission 4th Climate Change Assessment, Award EM4CD3-04A, and by NSF CyberSEES, Award 1539593. High-performance computing resources provided by the UC Davis College of Engineering

Source Language Python 2.7, 3.4

Supported Systems Unix, Linux, Windows, Mac

License MIT License

Availability https://github.com/ucd-cws/calvin

- *Cross-platform*: Model should run on Windows/OSX/Linux
- *Open data formats*: Input and output data should use only non-proprietary formats such as CSV and JSON
- *Freely available*: Programming language and solvers should be free and open-source. Several solvers described in this work are cost-free only for academic use, but they are not strictly required.
- *Separation of model and data*: The HOBBES project (https://hobbes.ucdavis.edu) stores the network dataset independent from any particular model, allowing this work to use multiple state-of-the-art solvers or models.

In the process of meeting these goals, a few specific features are added, as described in Section 2.

1.2. Network flow optimization

CALVIN is a network flow optimization model seeking to minimize statewide operating and water scarcity cost, subject to physical and regulatory constraints. Network flow problems commonly appear in energy and transportation problems in addition to water; references such as Bazaraa et al. (1977) provide detailed explanation and solution methods for network flow programming. The physical system is represented by a set of nodes $\mathscr{N}$ and links $\mathscr{A}$. Links are defined by $(i, j, k) \in \mathscr{A}$, where i is the source node, $j, j \neq i$ is the destination node, and k is the index of the piecewise linear component for links. Each link has the following properties: flow X_{ijk} (the decision variable), unit cost c_{ijk}, lower bound l_{ijk}, upper bound u_{ijk}, and amplitude (loss factor) a_{ijk}.

The objective function and constraints are:

$$\min_X z = \sum_i \sum_j \sum_k c_{ijk} X_{ijk} \tag{1}$$

subject to:

$$X_{ijk} \leq u_{ijk}, \forall\ (i, j, k) \in \mathscr{A} \tag{2}$$

$$X_{ijk} \geq l_{ijk}, \forall\ (i, j, k) \in \mathscr{A} \tag{3}$$

$$\sum_i \sum_k X_{jik} - \sum_i \sum_k a_{ijk} X_{ijk} = 0, \forall\ j \in \mathscr{N} \tag{4}$$

The objective function (Equation (1)) is a summation over all links i, j, k representing the total cost of flow conveyed in the network. The constraints (Equations (2)–(4)) enforce the upper and lower bounds on each link, and the mass balance at each node, respectively. The lower bound constraints include non-negativity: $l_{ijk} \geq 0\ \forall\ (i, j, k)$. The network flow formulation is a suitable model for large-scale water supply operations, but many other approaches have been proposed to model multi-reservoir systems (e.g., Labadie, 2004; Matrosov et al., 2011; Li et al., 2015; Giuliani et al., 2016).

The version of the model published by Draper et al. (2003) and used in subsequent studies employs the HEC-PRM solver created by the U.S. Army Corps of Engineers Hydrologic Engineering Center. This solver uses the out-of-kilter method for network flow optimization, and stores data in the binary HEC-DSS format. It is available for the Windows operating system. The CALVIN model requires approximately one week to solve using this software platform without an approximate initial solution.

2. Model description

2.1. California network dataset: HOBBES

To implement the model (Equations (1)–(4)), a set of nodes, links, and their properties must be provided. As part of this work, the database for California's water supply network has been migrated to the HOBBES platform (https://hobbes.ucdavis.edu). Unlike previous large-scale water resources optimization models, the HOBBES web-based framework separates the network data from the model-specific optimization method. This approach is intended to facilitate data standardization and documentation, increase transparency, reduce overhead development costs, and enhance modeling collaborations between academic, industry, government agencies, and non-governmental organizations (Medellín-Azuara et al., 2013). Similar benefits have been observed across a range of water resources web applications (Swain et al., 2015).

HOBBES combines a web visualization of the network (https://hobbes.ucdavis.edu/cwn) with the underlying database and metadata in JSON format (https://github.com/ucd-cws/calvin-network-data). The platform also includes automated scripts to export the database to a CSV format compatible with most solvers. The CALVIN GitHub repository provides links to CSV files that have been exported using this tool. Each row of the exported CSV file represents one link; the first two columns contain strings representing the names of the source and destination nodes i, j. The following columns contain the piecewise index k, cost c, amplitude a, lower bound l, and upper bound u.

The dataset currently covers the period 1922–2003 on a monthly timestep. Water scarcity costs are represented by piecewise linear functions for projected urban and agricultural demands in 2050 (Medellin-Azuara et al., 2007; Dogan et al., In Review). The physical network contains approximately 1000 spatial nodes and 60,000 links, selected types of which are shown in Fig. 1. The network model contains a copy of each node for each timestep, where links between timesteps represent storage at surface and groundwater storage nodes. Therefore, the size of the optimization problem increases linearly with the length of the model period. A 1-year model run contains approximately 60,000 decision variables (links), while the full 82-year run contains over 5 million decision variables, including piecewise linear water shortage and operating costs. These values also include optional "debug links" to address potential infeasibilities in the optimization, which are described in Section 2.3 and account for roughly 40% of the links in the model.

2.2. Dependencies and installation

The model requires the Python programming language and its standard scientific libraries (NumPy, SciPy, pandas); we recommend the Anaconda Distribution (https://www.continuum.io/downloads) to easily download and maintain these standard packages. Beyond this, the Pyomo library (Hart et al., 2012) is also required, and is not included with the standard packages in the Anaconda distribution. Pyomo

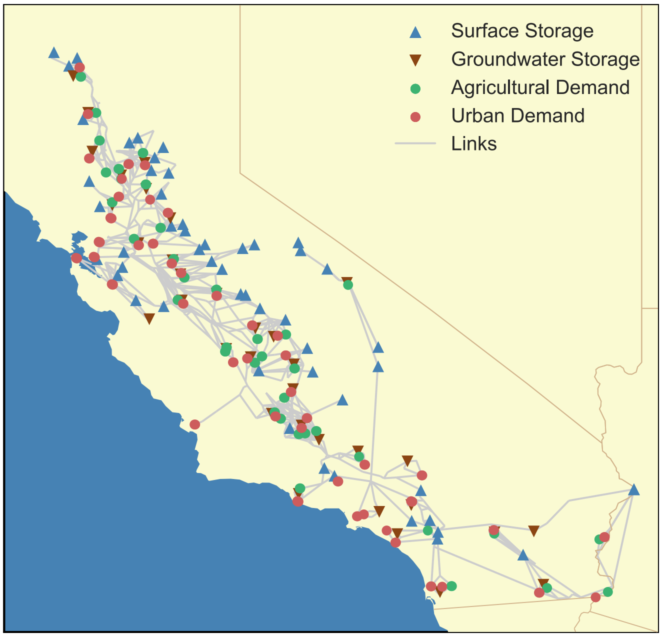


Fig. 1. Network representation of California's water supply system in the CALVIN model. Selected node types are shown; other node types include pumping plants, water treatment plants, and junctions.

provides a high-level interface for problem formulation that can be linked to different solvers, in the spirit of algebraic modeling languages such as GAMS and AMPL. Solvers must be installed separately. A simple method to install Pyomo along with the GLPK solver (Makhorin et al., 2008) using a command-line package manager is shown in the README file of this repository. In this short communication we also investigate the performance of the CBC (Forrest, 2012), Gurobi (2014), and CPLEX (IBM, 2009) solvers, all of which are free for academic use. GLPK and CBC are also free for commercial use. Installing these solvers and connecting to Pyomo must be done outside of the standard package managers, and we refer the reader to their respective documentation. The CALVIN model can be obtained by cloning or downloading the GitHub repository linked in the software availability section (hash a6a9fe7, May 2018). Links to download the network data in CSV format are provided in the README.

2.3. Resolving infeasibilities

The default model configurations should solve without issue, following the steps to be described in Section 2.5. However, when parameters are changed—for example, hydrologic inputs—the problem may become infeasible, which the solvers will quickly recognize. To locate and fix the constraints causing this problem, the package includes the option to run in “debug mode”. This is an automated version of a heuristic approach that has been developed in prior studies (Tanaka et al., 2006; Medellin-Azuara et al., 2007). Debug mode adds two additional nodes, “super-source” and “super-sink”, which are linked to all other nodes in the network at each timestep. These links can respectively add or remove as much water as needed for the infeasible constraints to become feasible, but at a very high cost—values ranging from $\$10^6$–$10^{10}$ per acre-foot have been used, orders of magnitude higher than any physically-based cost in the network. So, the solver will only allow flow on the debug links necessary for feasibility. The magnitude of these flows (typically small) is used to either decrease lower-bound constraints, increase upper-bound constraints, or otherwise identify input errors. The numerical tolerance to identify debug flows can be set by the user; the default tolerance is 10^{-7} acre-feet per month. The debug process also reports the total volume of constraint alterations when it is finished. Multiple iterations may be needed to eliminate all debug flows, after which the debug links are removed and the model can be solved. This debugging process increases the runtime substantially, because the model must be solved an additional 5–10 times. However, it is a straightforward way to obtain a feasible solution for a large model with a modified set of input parameters.

2.4. Perfect vs. limited foresight

When optimizing over decades-long time periods, the operation of surface and groundwater storage can be determined with perfect foresight of future dry and wet years. The limitations of this assumption have long been recognized (e.g., Newlin et al., 2002; Tanaka et al., 2006). Moving to a new software platform offers the flexibility to investigate the alternative assumption of limited foresight, where sequential annual optimizations apply the end-of-period storage from each year as the initial condition for the following year. The key challenge is defining optimization constraints or values for the end-of-period minimum carryover storage. To analyze the effect of carryover storage constraints, we vary them as a percentage of available surface reservoir capacity above dead pool, from 0% to 50% in steps of 5%. Groundwater storage volumes are constrained to the optimal end-of-year storage from the full 82-year run with perfect foresight. This assumption introduces more information into the limited foresight model, but prevents the limited foresight model from overdrafting groundwater to compensate for periods of low surface water availability. Further, it is a simplifying assumption that all of the large surface reservoirs would be operated for the same carryover targets as a percentage of their respective capacities; this may increase costs relative to individually specified carryover targets. More advanced methods to assign economic value to carryover storage at individual reservoirs are

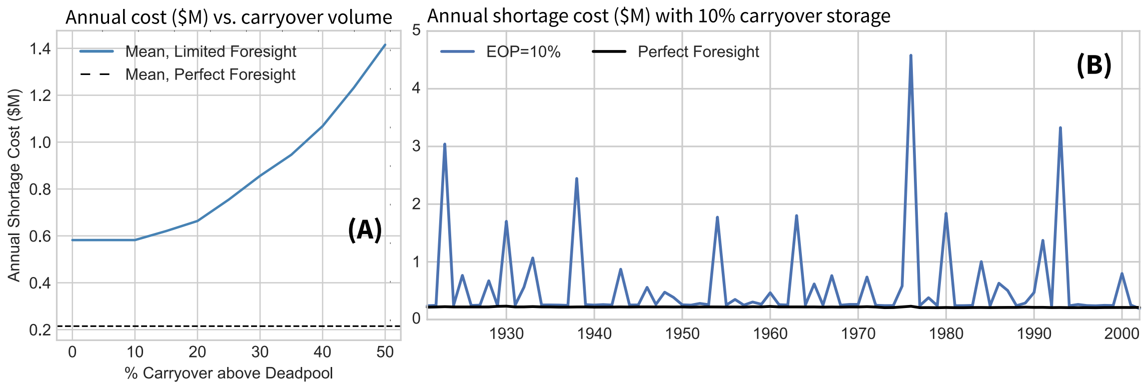


Fig. 2. (A) Average statewide annual cost as a function of the end-of-period storage volume in the 10 largest reservoirs, given in million acre-feet (MAF). (B) Timeseries of annual shortage cost with an end-of-period constraint of 10% of capacity above dead pool.

presented in Draper (2001); Draper and Lund (2004).

Fig. 2a shows that a minimum carryover storage constraint of about 5 million acre-feet (MAF), or roughly 10% above dead pool, results in the lowest average annual cost. The average annual shortage cost is only three times that in the perfect foresight case. Beyond that point, costs increase because storing too much for future years increases shortages in the current year. The timeseries in Fig. 2b shows that the limited foresight optimization is prone to spikes in annual shortage cost during drought years when they normally occur, whereas the perfect foresight model more evenly distributes cost over the period. Limited foresight reflects more realistic management, since accurate forecasts of drought events are not available years in advance. The annual timestep moves the model closer to a simulation model, but one in which network flows within each year are optimized assuming perfect foresight. We additionally expect the marginal value of storage capacity to increase under the limited foresight assumption, reflected in the dual values on the reservoir capacity constraints.

Limited foresight mode is a new addition to the CALVIN model, which has been enabled by the move to a new software platform. The impacts of limited foresight on water allocation and cost remain the subject of ongoing research. Section 2.5 contains an example of running connected annual optimizations in a loop.

2.5. Examples

Fig. 3 shows how to run a 1-year optimization for water year 1922 (beginning October 1921). The network data is imported from a CSV file. The model is first run in debug mode to eliminate infeasibilities. The results from Pyomo are postprocessed and stored in the directory example-results. Results include a set of eight CSV files: flow.csv, the flow on each link (TAF/month); storage.csv, surface and groundwater storage volumes at the end of each month (TAF); dual_lower.csv, dual_upper.csv, and dual_node.csv, which are respectively the dual values (shadow prices) on the lower/upper bound of each link, and the dual values on the mass balance constraint at each node. The final three files are shortage_volume.csv and shortage_cost.csv, given on selected links, and evaporation.csv, given at storage nodes. All results files contain 1 row for each month of the model run, and the columns are either link or node names as appropriate. The CSV output format allows the results to be analyzed and plotted using any standard software. This 1-year example solves with GLPK in approximately 1.5 min on a desktop computer, including two iterations of debug mode.

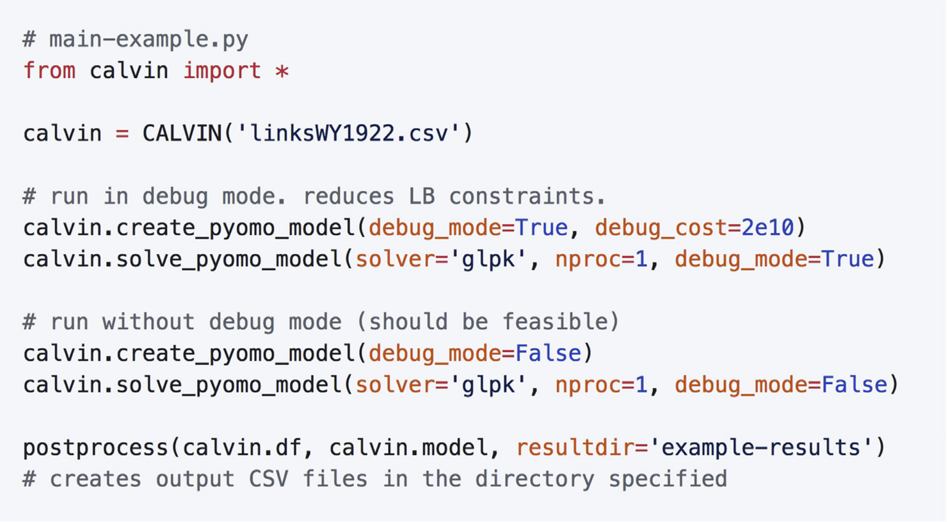
Fig. 4 shows how the limited foresight model can be run with multiple connected annual optimizations. At the start of each iteration, the model is created with initial conditions taken from the previous period's ending storage. The end-of-period constraint is set to 10% of available capacity above dead pool, as suggested by the results in Fig. 2. Each year is first run in debug mode, which may require several iterations, before solving for a feasible solution without debug mode. Results from each year are appended to the output CSV files. This example will require much more computing time than the 1-year run shown in Fig. 3; a rough estimate would be to multiply by the number of years, but this will not be exact due to differences in the number of debug mode iterations between years.

The examples in Figs. 3 and 4 would not require much modification to run different experiments. In fact, the most common changes, such as hydrologic inputs and network connections, would be made in the network links data file (CSV) and would not require any changes to the code. Advanced users seeking more information about the internal functions in the model should refer to the docstrings for each function on GitHub.

3. Runtime benchmarks

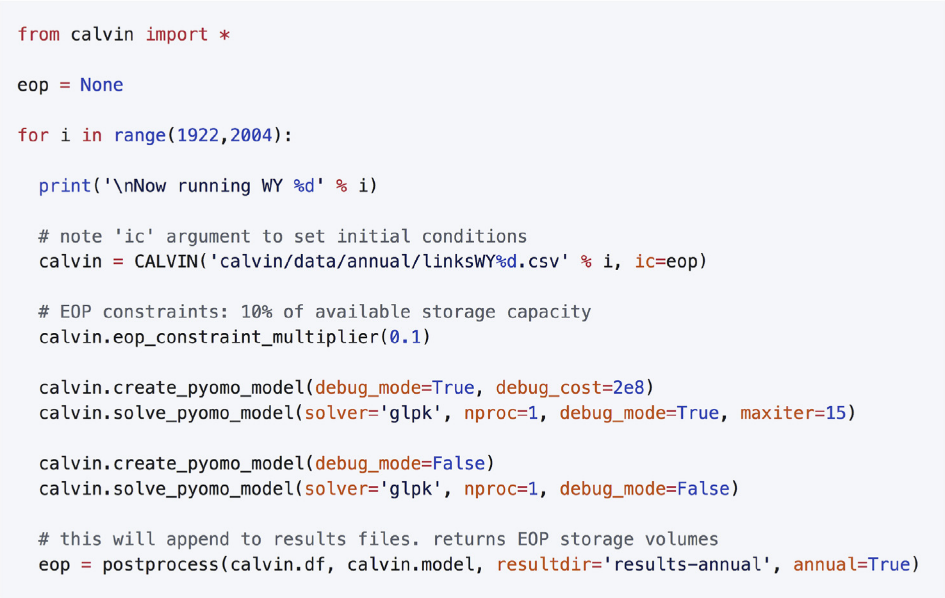
Several state-of-the-art linear programming solvers are available. It is useful to compare how runtimes increase with the number of decision variables. For this problem, the number of decision variables can be controlled by the number of years used in the optimization. Here we experiment with model runs of 1, 5, 10, 40, and 82 years, and record the solver runtime required in each case in debug mode, excluding time for file reading and writing. Only one debug iteration is needed, because these historical runs do not contain infeasibilities. Four solvers are tested (CBC, CPLEX, Gurobi, and GLPK), with 10 trials for each combination of solver and model size, for a total of 200 model runs. Tests are performed on the UC Davis HPC1 cluster, which contains 60 nodes each with 64 GB of RAM and two 8-core dual-threaded CPUs running at 2.4 GHz. The CPLEX, Gurobi, and CBC solvers use shared-memory parallelization on 32 threads, while GLPK runs in serial.

Fig. 5 shows solver runtimes with increasing numbers of decision variables. These results assume perfect foresight; runtime analysis of the limited foresight model remains a subject of ongoing work. Gurobi takes the least amount of time to find an optimal solution for all model sizes. Gurobi requires roughly 30 min to solve the largest model, with about five million decision variables in debug mode, while GLPK (serial) requires roughly 4.5 days. The speedup is partially a function of parallelization, but also the use of different techniques within each solver. The HEC-PRM solver used in the prior version of the model also runs in serial, and requires roughly 7 days to solve without an initial



```
# main-example.py
from calvin import *

calvin = CALVIN('linksWY1922.csv')

# run in debug mode. reduces LB constraints.
calvin.create_pyomo_model(debug_mode=True, debug_cost=2e10)
calvin.solve_pyomo_model(solver='glpk', nproc=1, debug_mode=True)

# run without debug mode (should be feasible)
calvin.create_pyomo_model(debug_mode=False)
calvin.solve_pyomo_model(solver='glpk', nproc=1, debug_mode=False)

postprocess(calvin.df, calvin.model, resultdir='example-results')
# creates output CSV files in the directory specified
```

Fig. 3. Example code for a 1-year optimization using water year 1922.



```
from calvin import *

eop = None

for i in range(1922,2004):

  print('\nNow running WY %d' % i)

  # note 'ic' argument to set initial conditions
  calvin = CALVIN('calvin/data/annual/linksWY%d.csv' % i, ic=eop)

  # EOP constraints: 10% of available storage capacity
  calvin.eop_constraint_multiplier(0.1)

  calvin.create_pyomo_model(debug_mode=True, debug_cost=2e8)
  calvin.solve_pyomo_model(solver='glpk', nproc=1, debug_mode=True, maxiter=15)

  calvin.create_pyomo_model(debug_mode=False)
  calvin.solve_pyomo_model(solver='glpk', nproc=1, debug_mode=False)

  # this will append to results files. returns EOP storage volumes
  eop = postprocess(calvin.df, calvin.model, resultdir='results-annual', annual=True)
```

Fig. 4. Example code for a limited foresight optimization using connected annual runs.

solution; GLPK shows a comparable order of magnitude.

As indicated by the regression lines in Fig. 5, solver solution times show a polynomial relationship with the number of decision variables (linear on a log-log scale). These relationships are all approximately quadratic, though Gurobi again displays the best result of $O(n^{1.55})$. Runtimes are consistent between trials, with only one outlier for CPLEX with the largest model size (a runtime of 1 day in Fig. 5). Because the solution method is deterministic, this outlier is likely due to a slowdown in either communication or file reading on the high-performance computing system rather than the solver itself. Further runtime improvements for all four solvers may be obtained by providing an initial solution based on previous results, a technique which has been used in previous studies. These results provide an application-focused runtime benchmark for large-scale network problems.

4. Conclusion

This short communication describes a new open-source implementation of the CALVIN model, a large-scale network flow optimization model of California's water supply system. The model is built on top of the Pyomo library and connects to multiple linear programming solvers that are freely available for academic use. The increased flexibility provided by this platform has enabled preliminary investigations into limited foresight optimization, as well as runtime benchmarks for different solvers on a high-performance computing cluster. The model runs on any operating system compatible with Python, and uses only non-proprietary data formats so that the input and output data can be easily analyzed. This software contribution is intended to facilitate future studies by researchers interested in economically-driven water resources allocation in California.

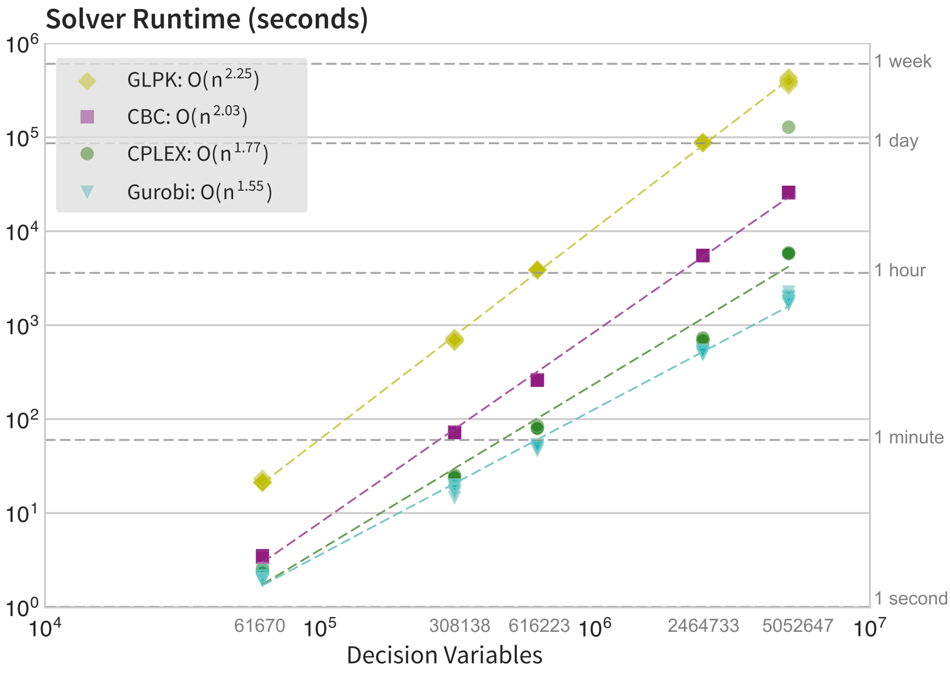


Fig. 5. Solver runtimes with linear trend lines and extrapolation equations on logarithmic scale. Runtimes do not include time for file reading and writing. This is wall clock time, not CPU time, so the single outlier point for CPLEX can be attributed to I/O and communication slowdown on the HPC system rather than to the solver itself.

Acknowledgments

This work was partially supported by the California Energy Commission 4th Climate Change Assessment, Award EM4CD3-04A, and by NSF CyberSEES, Award 1539593. High-performance computing resources were provided by the UC Davis College of Engineering.

References

Bazaraa, M.S., Jarvis, J.J., Sherali, H.D., 1977. Linear Programming and Network Flows, second ed. John Wiley & Sons.

Booker, J.F., Howitt, R.E., Michelsen, A.M., Young, R.A., 2012. Economics and the modeling of water resources and policies. Nat. Resour. Model. 25, 168–218. https://doi.org/10.1111/j.1939-7445.2011.00105.x.

Cai, X., 2008. Implementation of holistic water resources-economic optimization models for river basin management reflective experiences. Environ. Model. Software 23, 2–18. https://doi.org/10.1016/j.envsoft.2007.03.005. URL: http://www.sciencedirect.com/science/article/pii/S1364815207000746.

Dogan, M., Buck, I., Lund, J., 2018. Statewide effects of ending groundwater overdraft in California. J. Water Resour. Plann. Manag (In Review).

Dogrul, E.C., Kadir, T.N., Brush, C.F., Chung, F.I., 2016. Linking groundwater simulation and reservoir system analysis models: the case for californias central valley. Environ. Model. Software 77, 168–182. https://doi.org/10.1016/j.envsoft.2015.12.006.

Draper, A.J., 2001. Implicit Stochastic Optimization with Limited Foresight for Reservoir Systems. Ph.D. thesis University of California, Davis PhD dissertation. .

Draper, A.J., Jenkins, M.W., Kirby, K.W., Lund, J.R., Howitt, R.E., 2003. Economic-engineering optimization for California water management. J. Water Resour. Plann. Manag. 129, 155–164. https://doi.org/10.1061/(ASCE)0733-9496(2003)129:3(155).

Draper, A.J., Lund, J.R., 2004. Optimal hedging and carryover storage value. J. Water Resour. Plann. Manag. 130, 83–87.

Forrest, J., 2012. CBC (COIN-OR Branch and Cut) Open-source Mixed Integer Programming Solver. URL. https://projects.coin-or.org/Cbc.

Giuliani, M., Li, Y., Cominola, A., Denaro, S., Mason, E., Castelletti, A., 2016. A matlab toolbox for designing multi-objective optimal operations of water reservoir systems. Environ. Model. Software 85, 293–298. https://doi.org/10.1016/j.envsoft.2016.08.015.

Gurobi, 2014. Gurobi Optimizer Reference Manual. URL: http://www.gurobi.com.

Harou, J.J., Pulido-Velazquez, M., Rosenberg, D.E., Medellín-Azuara, J., Lund, J.R., Howitt, R.E., 2009. Hydro-economic models: concepts, design, applications, and future prospects. J. Hydrol. 375, 627–643. https://doi.org/10.1016/j.jhydrol.2009.06.037.

Hart, W.E., Laird, C., Watson, J.-P., Woodruff, D.L., 2012. Pyomo Optimization Modeling in Python Volume 67 of Springer Optimization and its Applications. Springer US, Boston, MA. https://doi.org/10.1007/978-1-4614-3226-5.

Howitt, R.E., Medellín-Azuara, J., MacEwan, D., Lund, J.R., 2012. Calibrating disaggregate economic models of agricultural production and water management. Environ. Model. Software 38, 244–258. https://doi.org/10.1016/j.envsoft.2012.06.013.

IBM, 2009. V12.1: users manual for CPLEX. vol. 46. International Business Machines Corporation, pp. 157.

Labadie, J.W., 2004. Optimal operation of multireservoir systems: state-of-the-art review. J. Water Resour. Plann. Manag. 130, 93–111. https://doi.org/10.1061/(ASCE)0733-9496(2004)130:2(93).

Li, F.-F., Shoemaker, C.A., Qiu, J., Wei, J.-H., 2015. Hierarchical multi-reservoir optimization modeling for real-world complexity with application to the three gorges system. Environ. Model. Software 69, 319–329. https://doi.org/10.1016/j.envsoft.2014.11.030. URL: http://www.sciencedirect.com/science/article/pii/S1364815214003612.

Makhorin, A., et al., 2008. GLPK (GNU Linear Programming Kit).

Matrosov, E.S., Harou, J.J., Loucks, D.P., 2011. A computationally efficient open-source water resource system simulator application to london and the thames basin. Environ. Model. Software 26, 1599–1610. https://doi.org/10.1016/j.envsoft.2011.07.013.

Medellin-Azuara, J., Harou, J.J., Olivares, M.A., Madani, K., Lund, J.R., Howitt, R.E., Tanaka, S.K., Jenkins, M.W., Zhu, T., 2007. Adaptability and adaptations of California's water supply system to dry climate warming. Climatic Change 87. https://doi.org/10.1007/s10584-007-9355-z.

Medellín-Azuara, J., Sandoval Solis, S., Lund, J., Chu, W., 2013. Building models from the bottom up: the HOBBES project. In: AGU Fall Meeting Abstracts.

Newlin, B.D., Jenkins, M.W., Lund, J.R., Howitt, R.E., 2002. Southern California water markets: potential and limitations. J. Water Resour. Plann. Manag. 128, 21–32.

Quinn, N., Brekke, L., Miller, N., Heinzer, T., Hidalgo, H., Dracup, J., 2004. Model integration for assessing future hydroclimate impacts on water resources, agricultural production and environmental quality in the san joaquin basin, California. Environ. Model. Software 19, 305–316. Concepts, Methods and Applications in Environmental Model Integration. https://doi.org/10.1016/S1364-8152(03)00155-5.

Swain, N.R., Latu, K., Christensen, S.D., Jones, N.L., Nelson, E.J., Ames, D.P., Williams, G.P., 2015. A review of open source software solutions for developing water resources web applications. Environ. Model. Software 67, 108–117. https://doi.org/10.1016/j.envsoft.2015.01.014.

Tanaka, S.K., Zhu, T., Lund, J.R., Howitt, R.E., Jenkins, M.W., Pulido, M.A., Tauber, M., Ritzema, R.S., Ferreira, I.C., 2006. Climate warming and water management adaptation for California. Climatic Change 76, 361–387. https://doi.org/10.1007/s10584-006-9079-5.

Winter, J.M., Young, C.A., Mehta, V.K., Ruane, A.C., Azarderakhsh, M., Davitt, A., McDonald, K., Haden, V.R., Rosenzweig, C., 2017. Integrating water supply constraints into irrigated agricultural simulations of California. Environ. Model. Software 96, 335–346. https://doi.org/10.1016/j.envsoft.2017.06.048.

Yang, T., Gao, X., Sellars, S.L., Sorooshian, S., 2015. Improving the multi-objective evolutionary optimization algorithm for hydropower reservoir operations in the California Oroville-Thermalito complex. Environ. Model. Software 69, 262–279. https://doi.org/10.1016/j.envsoft.2014.11.016. URL: http://www.sciencedirect.com/science/article/pii/S1364815214003429.