Mooflow: A generalized framework for basin-scale multi-objective simulation-optimization with MODFLOW

Key findings

- We introduce the software package Mooflow for Python to set up and perform generalized multi-objective simulation-optimization of groundwater management.
- Mooflow couples the groundwater model MODFLOW with the water distribution network model Pywr and parallelized heuristic optimization algorithms.
- Application of the framework is demonstrated on a fictional water resource system for the Western Mountain Aquifer.
- The software will be made publicly available as open source at https://bitbucket.org/BAH_Berlin/mooflow.

Motivation

Sustainable water resources management (WRM) in arid regions is often subject to conflicting interests. A typical problem in groundwater-related WRM is the optimization of groundwater extraction rates. The objectives to maximize sustainability or other ecological aspects conflict with the minimization of pumping costs or shortages.

Multi-objective simulation-optimization (MOSO) provides water managers with a diverse set of best possible (Pareto-optimal) compromise solutions regarding these conflicting interests. In MOSO, a heuristic optimization algorithm is coupled to a groundwater model to evaluate the aquifer response. Additionally, for basin-wide optimization tasks and conjunctive use management, the integration of a superordinate branching water distribution system can be beneficial for modeling. Currently, no software is available to provide a generalized framework to help users set up and run a MOSO with MODFLOW (Harbaugh, 2005). Therefore, we introduce Mooflow for the Python programming language to integrate MODFLOW with the Pywr water distribution model (Tomlinson et al., 2020) and state-of-the-art heuristic optimization algorithms.

Methodology

Basin-wide optimization can comprise many wells and a substantial number of decision variables. To reduce optimization complexity, Mooflow features methods to minimize the number of decision variables by grouping wells and sharing decision variables. Because the computing time of basin-wide groundwater models can be extensive, parallelization may be necessary. The Python package DEAP (Fortin et al., 2012) provides powerful multi-objective algorithms and parallelization.

Mooflow uses an object-oriented design to be general, modular, and expandable. A variety of classes and functions are provided for a step-by-step setup of a MOSO:

1. An optimization algorithm varies extraction or recharge rates and evaluates the aquifer response.
2. Groundwater levels, drainage, and extractions are read from MODFLOW. Total extractions and recharge are assigned to groundwater source nodes or recharge nodes in the Pywr-model.
3. Pywr is executed to distribute the water available in the system for each time step or as a total balance.
4. Fitness function values are computed and returned to the optimization algorithm.

Mooflow

Setting up a simulation-optimization requires the development of code to modify configuration of models, run models, read model ouputs, and compute fitness functions. A generalized framework provides functionality to solve these complicated tasks for a variety of problems with a few lines of code. Mooflow leverages Python, an easy-to-use high-level programming language, hugely popular in science and engineering. It adds to an already extensive library of Python packages.

MedWater – Sustainable management of politically and economically highly relevant water resources in hydraulically, climatically, and ecologically dynamic carbonate aquifers of the Mediterranean.

Funding code: 02WGR1428A

SPONSORED BY THE Federal Ministry of Education and Research
FONA Research for Sustainable Development BMBF
An Initiative of the Federal Ministry of Education and Research GRoW WATER AS A GLOBAL RESOURCE

Mooflow: A generalized framework for basin-scale multi-objective simulation-optimization with MODFLOW

Results

Mooflow is applied to the Western Mountain Aquifer (WMA) in Israel and the West Bank for a scenario in the year 2040 with a 17% mean reduction in groundwater recharge. A total of 513 extraction wells and 32 recharge wells with seasonal extraction/recharge patterns are integrated in a MODFLOW model of the WMA. The seasonal extraction/recharge patterns are scaled up or down using ten decision variables that are subject to optimization. Figure 1 shows the superordinate fictional branched water distribution network that is simulated with Pywr. All wells are assigned to different management parcels and provide water to, or receive water from, the respective GW_ nodes in the water distribution network. Each parcel further contains additional demand and source nodes. Three fitness functions describe the management goals of increasing sustainability, decreasing provision costs, and reducing overall shortages in water supply until 2040. Figure 2 shows the approximated Pareto-Front after 2000 model evaluations. Solution S0 provides a sustainable management, cutting back slightly on groundwater extraction and allowing for minor shortages in water supply. The water footprint of imported crops increases as fewer crops are produced in the WMA, because less irrigation water from groundwater sources is available. The increase in imports and supplied water from other sources result in rising costs.

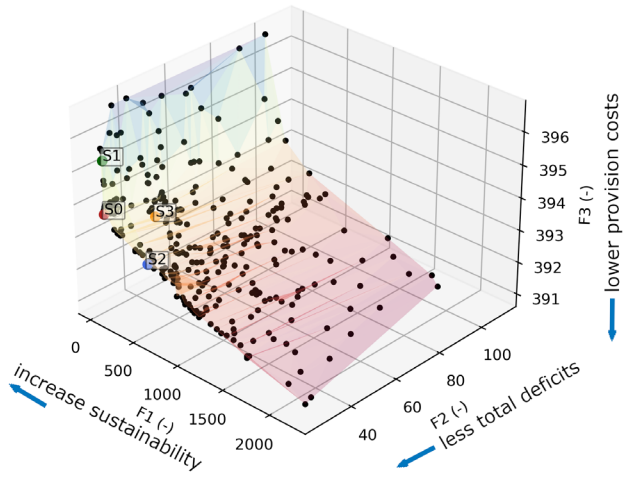

Figure 2: Pareto-Front with four selected solutions S0 to S3

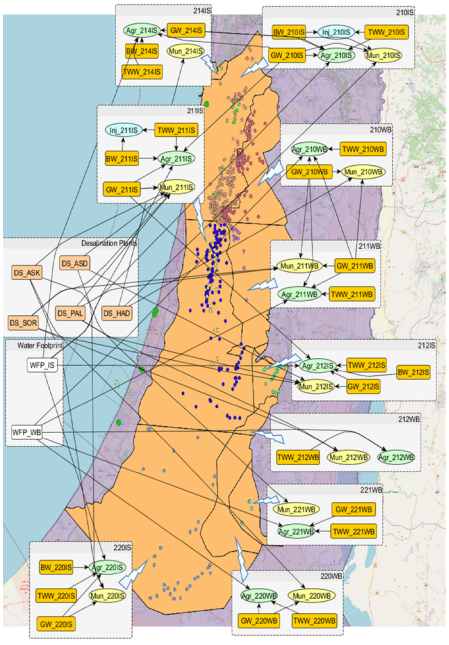

Figure 1: Water distribution system with 10 management sections. Source nodes are TWW – treated wastewater, GW – groundwater, BW – brackish water, DS – desalination plants, and WFP – water footprint of imports. Demand nodes are Munmunicipal/industrial and Agricultural. Arrows show flow directions. Points depict the extraction wells, except green in 210IS and light brown in 211IS, which are recharge wells.

Application

The software package Mooflow for Python enables users to set up and perform multi-objective simulation-optimization of groundwater management with MODFLOW. The integration of the Pywr water distribution model allows to integrate the flexibility of the superordinate water network to distribute water from different sources to demands and can prevent limitations in the design space of the optimization. Further development will extend the capabilities by integrating the LAK3 model for better conjunctive use management modeling.

References

Fortin, F.A., De Rainville F.M., Gardner, M.A., Parizeau, M., &Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research, 13*, 2171-2175. https://www.jmlr.org/papers/volume13/fortin12a/fortin12a.pdf

Tomlinson, J.E., Arnott, J.H., &Harou, J.J. (2020). A water resource simulator in Python. *Environmental Modelling & Software, 126*, 104653. https://doi.org/10.1016/j.envsoft.2020.104635

Harbaugh, A.W. (2005) MODFLOW-2005: The U.S. Geological Survey modular ground-water model: The ground-water flow process. U.S. Geological Survey. https://doi.org/10.3133/tm6A16

Author
Ruben Müller

Affiliation
Büro für Angewandte Hydrologie

Email
ruben.mueller@bah-berlin.de