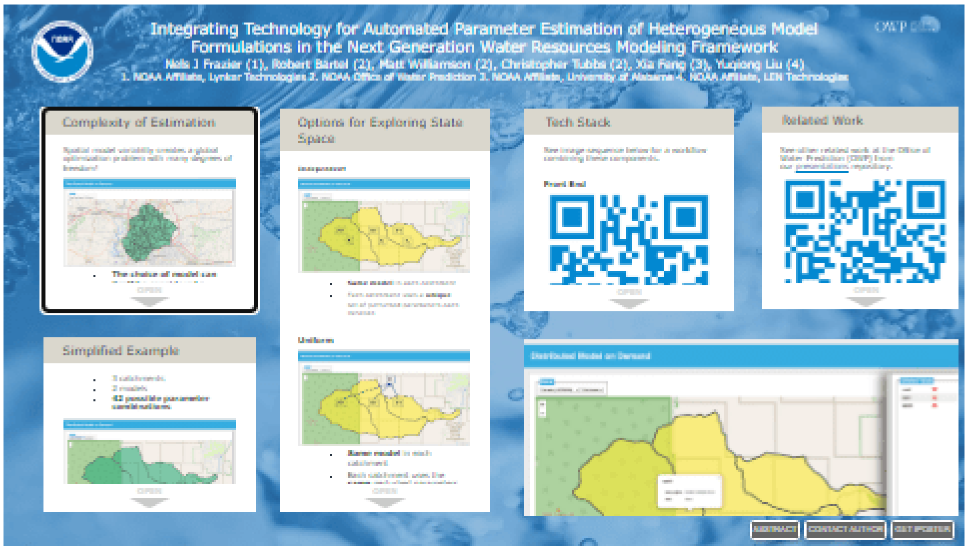
Integrating Technology for Automated Parameter Estimation of Heterogeneous Model Formulations in the Next Generation Water Resources Modeling Framework



Nels J Frazier (1), Robert Bartel (2), Matt Williamson (2), Christopher Tubbs (2), Xia Feng (3), Yuqiong Liu (4)

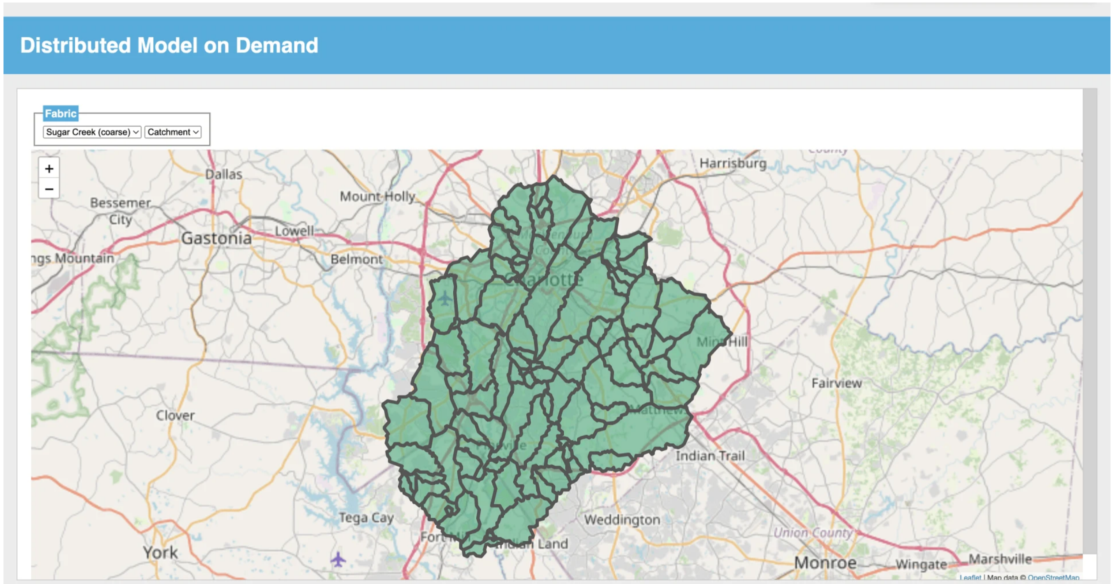
1. NOAA Affiliate, Lynker Technologies 2. NOAA Office of Water Prediction 3. NOAA Affiliate, University of Alabama 4. NOAA Affiliate, LEN Technologies

PRESENTED AT:

Frontiers in Hydrology

COMPLEXITY OF ESTIMATION

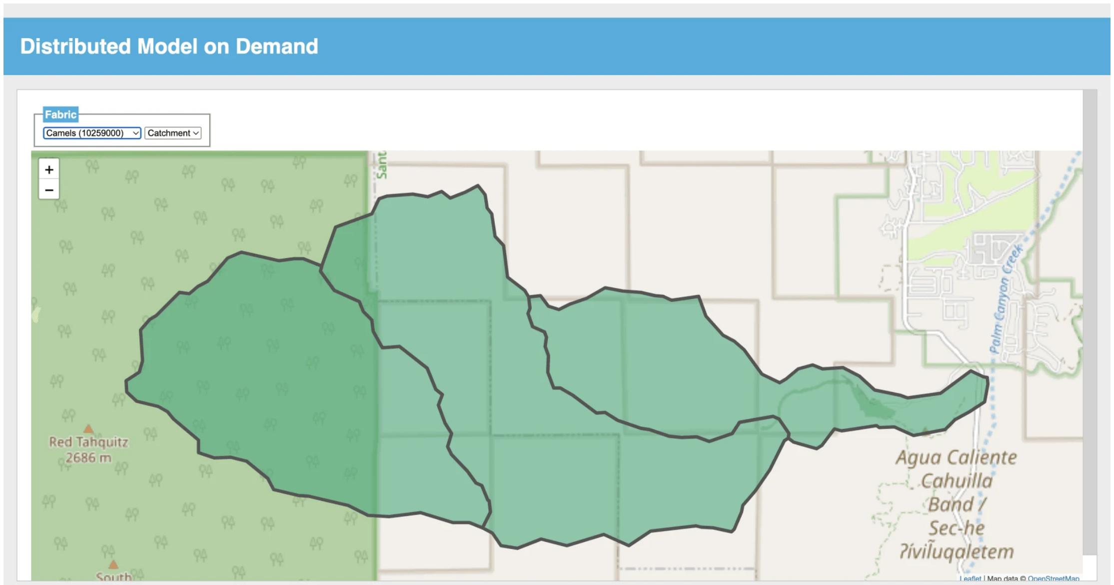
Spatial model variablity creates a global optimization problem with many degrees of freedom!



- **The choice of model can itself be considered a parameter**
- Each catchment has its own model
- Each model has its own parameter space

SIMPLIFIED EXAMPLE

- 3 catchments
- 2 models
- **42 possible parameter combinations**



The two models currently available for estimation are:

- CFE (Conceptual Functional Equivalent)
  - 9 estimatable parameters
- Topmodel
  - 5 estimatable parameters

Let the set of CFE parameters be C and the set of Topmodel parameters be T.

If we use a single model and parameter space over the entire domain, then the maximum number of parameters to estimate is $max(|T|, |C|)$. We call this a Uniform calibration.

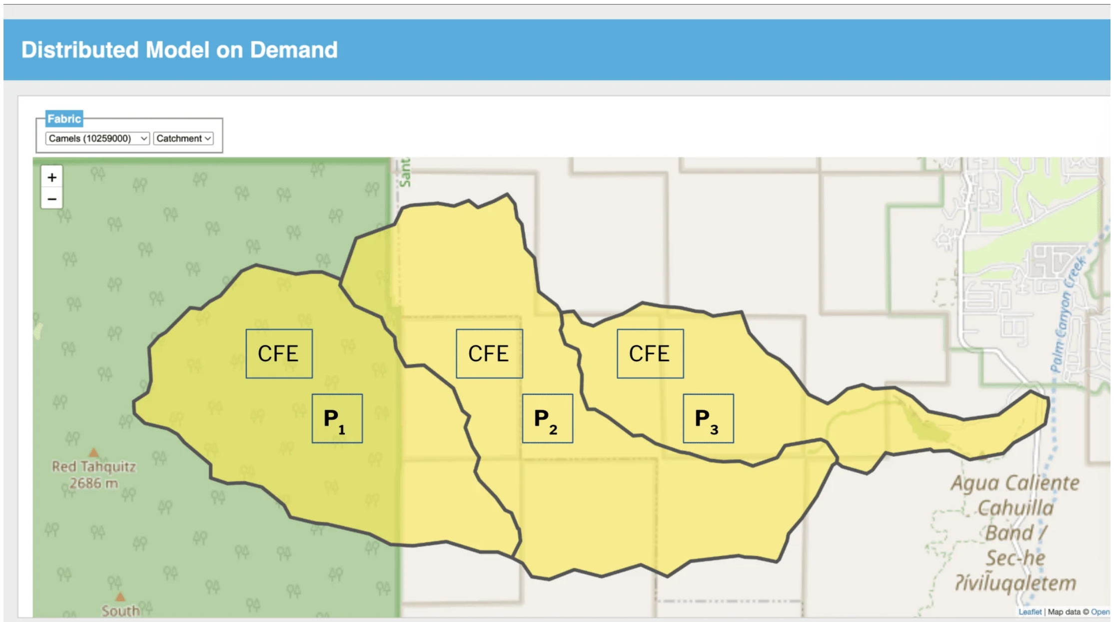
If we allow each catchment to have its own parameter space, but limited to a single model in each catchment, then the maximum number of parameters to estimate is $N * max(|T|, |C|)$ where N is the number of catchments we are calibrating. We call this an Independent calibration.

If we allow the model selection to be an estimatable parameter, we now have $N * (|T| + |C|)$ possible parameters to explore in our global search space!

For this small example, that is $3 * (5 + 9) = 3 * 14 = 42$ degrees of freedom!

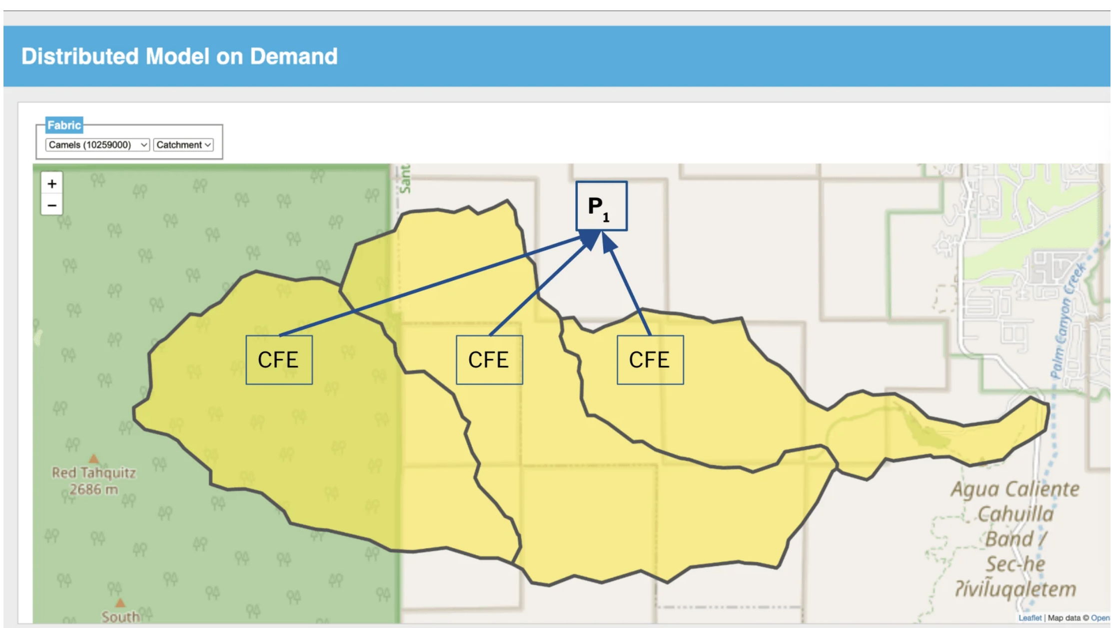
OPTIONS FOR EXPLORING STATE SPACE

Independent



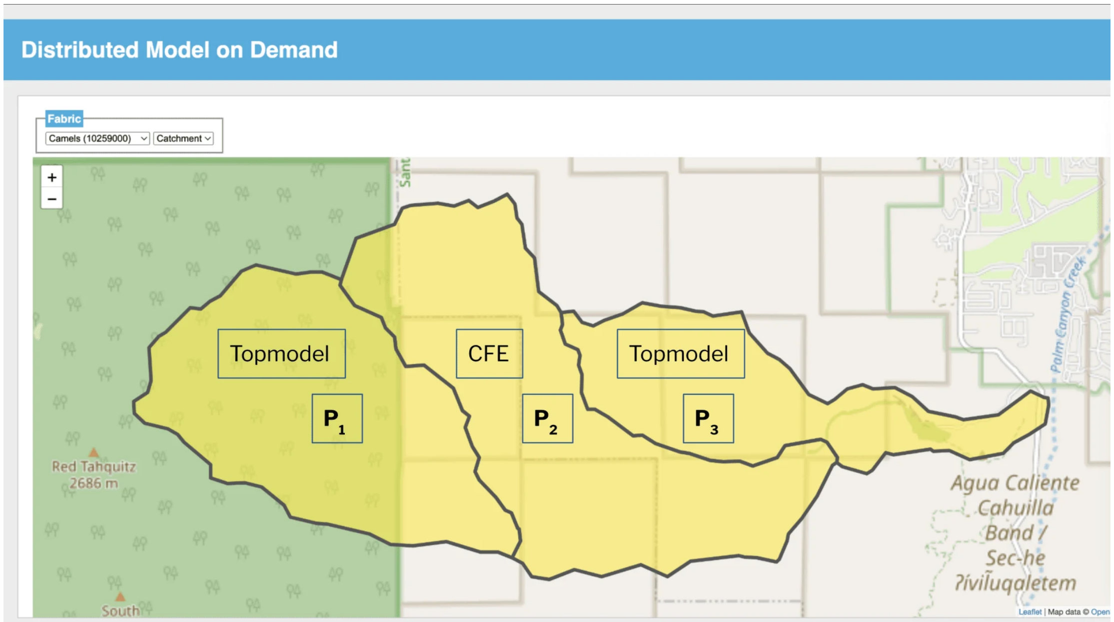
- **Same model** in each catchment
- Each catchment uses a **unique** set of perturbed parameters each iteration

Uniform



- **Same model** in each catchment
- Each catchment uses the **same** perturbed parameters each iteration

Explicit



- User defined, each catchment can be given a **unique model** and a **unique** parameter set

TECH STACK

See image sequence below for a workflow combining these components.

Front End

- Office of Water Prediction (OWP) Model as a Service using Distributed Model on Demand (DMOD) (https://github.com/noaa-owp/dmod)
    - Collects user input, prepares data (hydrofabric, forcing, ect), and connects to appropriate framework
    - Docker, Django, React

Middleware

- Parameter estimation package ngen-cal (https://github.com/noaa-owp/ngen-cal)
    - Wrapped with DMOD communication service (In Progress)
    - Creates framework input file based on hydrofabric and user options
    - Executes model, evaluates output, manages parameter space and estimation loop
    - Provides results to user (In Progress)
    - Docker, Python

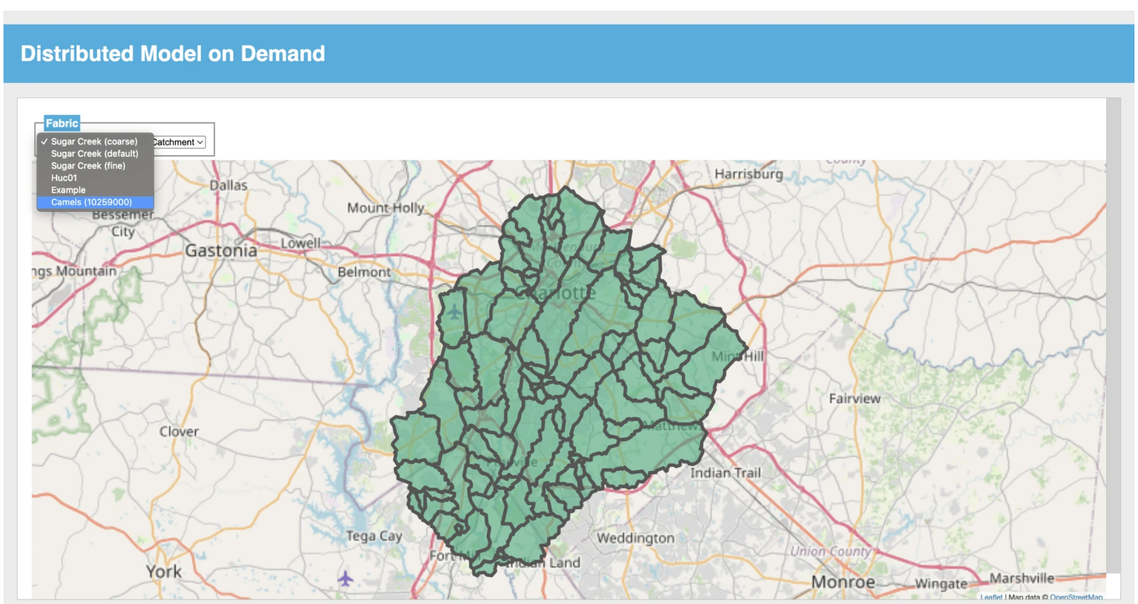
Backend

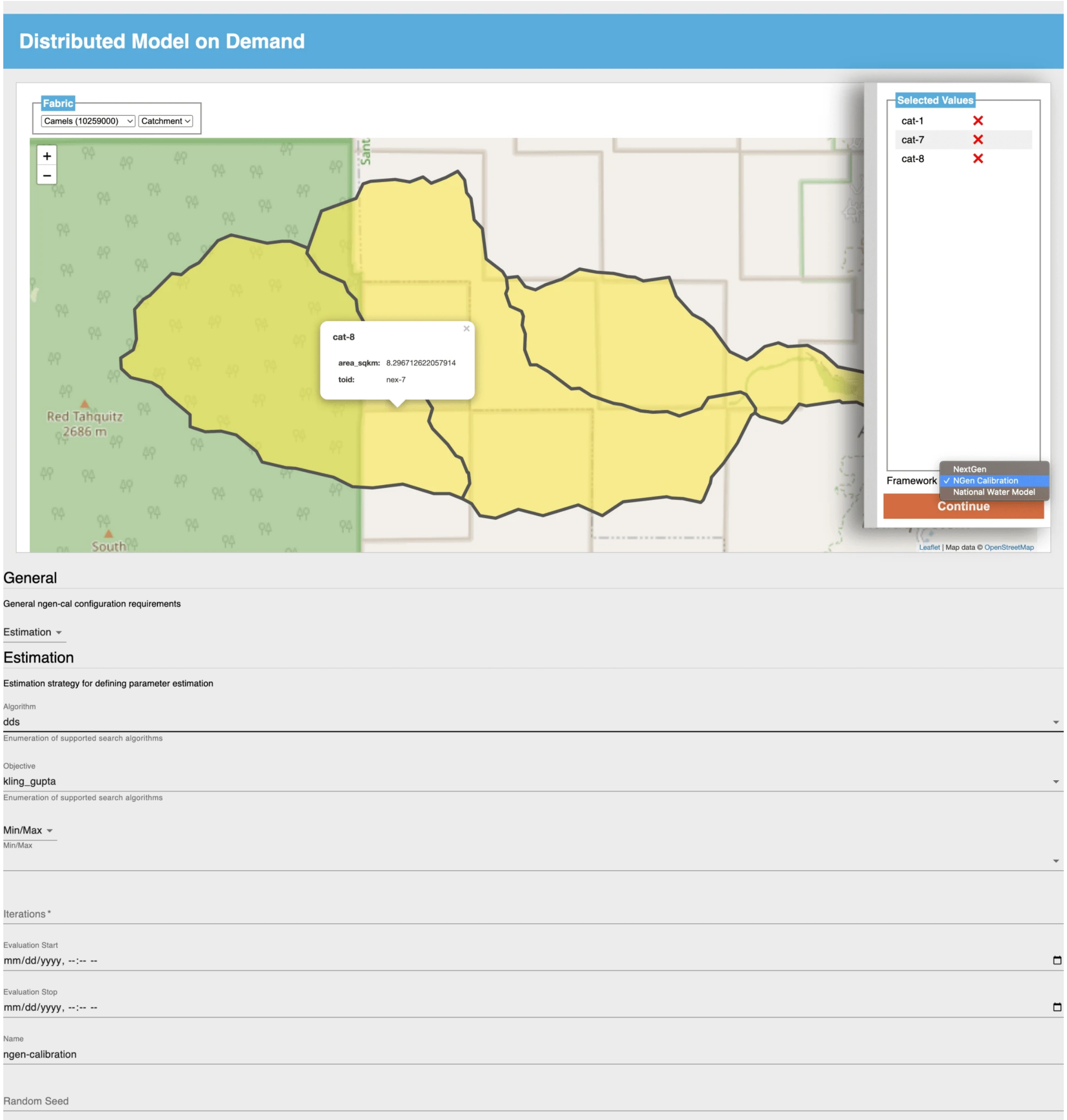
- Next Generation Water Resources Modeling Framework ngen (https://github.com/noaa-owp/ngen)
- Hydrofabric (https://github.com/NOAA-OWP/hydrofabric)
- Basic Model Interface (BMI (https://github.com/csdms/bmi))
- BMI Models
    - Conceptual Functional Equivalent (CFE (https://github.com/noaa-owp/cfe))
    - Topmodel (https://github.com/noaa-owp/topmodel/)
- C++, C, Fortran, Python, MPI

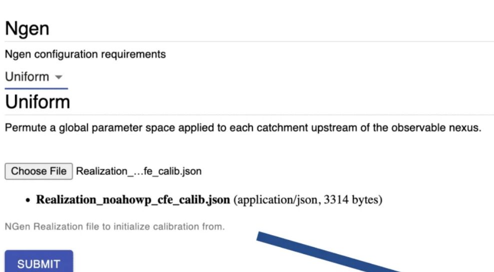
RELATED WORK

See other related work at the Office of Water Prediction (OWP) from our presentations (https://github.com/NOAA-OWP/OWP-Presentations) repository.

Or browse all of OWP's open source projects under our GitHub Organization (https://github.com/noaa-owp/)



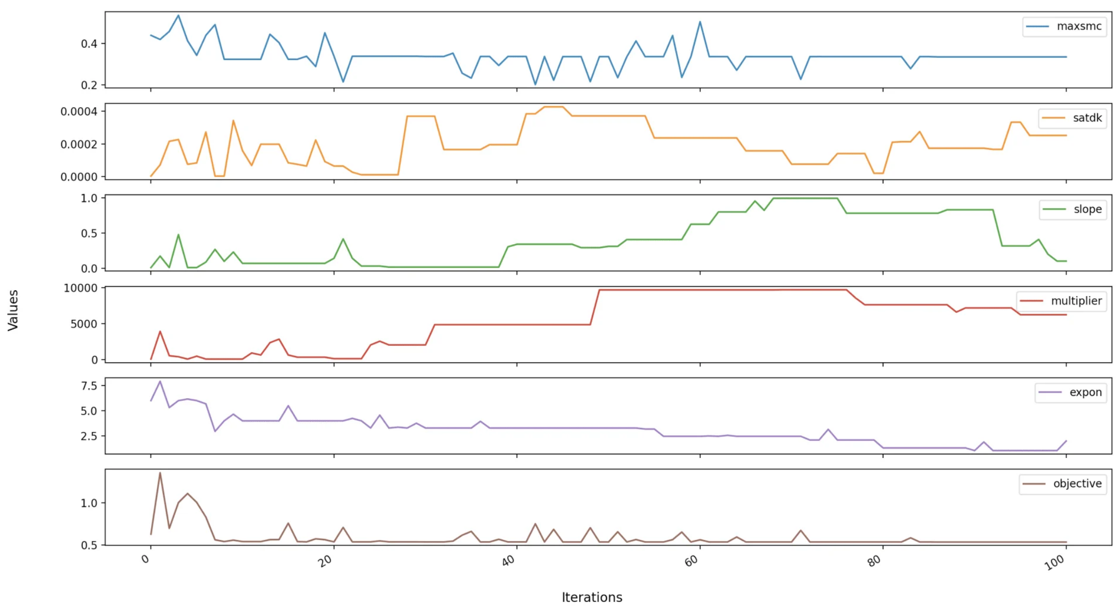





```
#Model specifc configuration
model:
    # Which model to execute for the search optimization
    # Currently only support `ngen` and `none` (for testing purposes)
    type: ngen
    # A binary in $PATH or a qualified path to the binary to run
    binary: "ngen"
    # If running ngen in parallel, provide the number of requested parallel processes
    # The binary will get prepended with mpirun -n `parallel` automatically
    # Must be >= 2
    #parallel: 2
    # If a parallel run is requested, ngen requires a static partitions.json input file
    # this is the path to that file
    #partitions: <path>
    # By default, ngen args will be `catchments`` "all" `nexus`` "all" `configuration`
    # and if running in parallel    `catchments`` "all" `nexus`` "all" `configuration` `partitions`
    # If you provide a custom arg string here, these are passed directly to the binary
    # and no adjustments are made
    #args: null
    # Required path to ngen realization config (with calibration info included)
    realization: ./Realization_noahowp_cfe_calib.json
    # Required path to catchment hydrofabirc file
    catchments: ./10259000/spatial/catchment_data.geojson
    # Required path to nexus hydrofabric file
    nexus: ./10259000/spatial/nexus_data.geojson
    # Required path to hydrofabric crosswalk file
    crosswalk: ./10259000/parameters/cross-walk.json
    #ngen calibration strategies include
    #multi: <describe multi strategey>
    #explicit: only calibrates basins in the realization_config with a "calibration" definition
    strategy: uniform
```

```
Starting calib
Starting Iteration: 0
Starting Best param: 0
Starting Best score: inf
Starting DDS loop
Running ngen spatial/catchment_data.geojson "all" spatial/nexus_data.geojson "all" Realization_noahowp_cfe_calib.json to produce initial simulation
Current score 0.6282143983385257
Best score 0.6282143983385257
Best parameters at iteration 0
inclusion probability: 1.0
neighborhood: 1    1
4    4
2    2
0    0
3    3
dtype: int64
Running ngen spatial/catchment_data.geojson "all" spatial/nexus_data.geojson "all" Realization_noahowp_cfe_calib.json for iteration 1
Current score 1.356096219750771
Best score 0.6282143983385257
Best parameters at iteration 0
inclusion probability: 0.8494850021680094
neighborhood: 4    4
3    3
0    0
1    1
dtype: int64
Running ngen spatial/catchment_data.geojson "all" spatial/nexus_data.geojson "all" Realization_noahowp_cfe_calib.json for iteration 2
Current score 0.6963571053276086
Best score 0.6282143983385257
Best parameters at iteration 0
inclusion probability: 0.7614393726401688
neighborhood: 1    1
2    2
0    0
3    3
dtype: int64
Running ngen spatial/catchment_data.geojson "all" spatial/nexus_data.geojson "all" Realization_noahowp_cfe_calib.json for iteration 3
Current score 1.0017261806287932
Best score 0.6282143983385257
Best parameters at iteration 0
inclusion probability: 0.6989700043360189
neighborhood: 0    0
```

Search Space



ABSTRACT

Parameter estimation at scale is a challenging problem because direct calibration of each catchment within an entire large-scale domain is not practical. The Next Generation Water Resources Modeling Framework (NextGen) enables the user to apply a different model formulation at each catchment within a larger domain. When multiple model formulations require parameter estimation at these scales, the problem complexity is increased astronomically. Questions remain unanswered about how the parameter space of multiple formulations interact, and how to distribute these calibrated parameters across the domain. Addressing these questions requires executing and evaluating the various model configurations for various permutations many times. In turn, this requires careful management of the data in and outputs from these executions. The NextGen design provides great flexibility in terms of hydrologic model configuration and execution. By itself, the framework brings computational science to bear in the initialization and orchestration of hydrologic models across physical domains of various scales using the WaterML version 2.0 HY_Features data model and the Basic Model Interface (BMI), key technologies that facilitate the management and flow of geospatial hydrologic data and model coupling, respectively. This enables the framework to select, configure, and apply model formulations at different scales, providing fine-grained control over model execution. Building on the BMI concept, we implement another technologic layer that unlocks specific internal model components required for automated parameter estimation. Alongside these core framework technologies, we developed parameter estimation capabilities linked to the framework using the common HY_Features data model. We present an integration of these key software technologies within the Distributed Model on Demand infrastructure utility. We show how all these pieces are combined together to bring automated calibration of Next Generation National Water Model formulations to users in a convenient and efficient manner.