Integrated High-Resolution Modeling for Operational Hydrologic Forecasting

by

Felipe Hernández

Bachelor of Science in Civil Engineering, Universidad de los Andes, 2008

Master of Science in Computer Engineering, Universidad de los Andes, 2010

Submitted to the Graduate Faculty of

Swanson School of Engineering in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

University of Pittsburgh

2019

UNIVERSITY OF PITTSBURGH

SWANSON SCHOOL OF ENGINEERING

This dissertation was presented

by

Felipe Hernández

It was defended on

March 21, 2019

and approved by

Jeen-Shang Lin, Ph.D., Associate Professor, Department of Civil and Environmental Engineering

Zhi-Hong Mao, Ph.D., Professor, Department of Electrical and Computer Engineering

Luis Vallejo, Ph.D., Professor, Department of Civil and Environmental Engineering

Dissertation Director: Xu Liang, Ph.D., Professor, Department of Civil and Environmental Engineering

Copyright © by Felipe Hernández

2019

Integrated High-Resolution Modeling for Operational Hydrologic Forecasting

Felipe Hernández, PhD

University of Pittsburgh, 2019

Current advances in Earth-sensing technologies, physically-based modeling, and computational processing, offer the promise of a major revolution in hydrologic forecasting—with profound implications for the management of water resources and protection from related disasters. However, access to the necessary capabilities for managing information from heterogeneous sources, and for its deployment in robust-enough modeling engines, remains the province of large governmental agencies. Moreover, even within this type of centralized operations, success is still challenged by the sheer computational complexity associated with overcoming uncertainty in the estimation of parameters and initial conditions in large-scale or high-resolution models.

In this dissertation we seek to facilitate the access to hydrometeorological data products from various U.S. agencies and to advanced watershed modeling tools through the implementation of a lightweight GIS-based software package. Accessible data products currently include gauge, radar, and satellite precipitation; stream discharge; distributed soil moisture and snow cover; and multi-resolution weather forecasts. Additionally, we introduce a suite of open-source methods aimed at the efficient parameterization and initialization of complex geophysical models in contexts of high uncertainty, scarce information, and limited computational resources. The developed products in this suite include: 1) model calibration based on state of the art ensemble evolutionary Pareto optimization, 2) automatic parameter estimation boosted through the incorporation of expert criteria, 3) data assimilation that hybridizes particle smoothing and variational strategies, 4) model state compression by means of optimized clustering, 5) high-dimensional stochastic approximation of watershed conditions through a novel lightweight Gaussian graphical model, and 6) simultaneous estimation of model parameters and states for hydrologic forecasting applications.

Each of these methods was tested using established distributed physically-based hydrologic modeling engines (VIC and the DHSVM) that were applied to watersheds in the U.S. of different sizes—from a small highly-instrumented catchment in Pennsylvania, to the basin of the Blue River in Oklahoma. A series of experiments was able to demonstrate statistically-significant improvements in the predictive accuracy of the proposed methods in contrast with traditional approaches. Taken together, these accessible and efficient tools can therefore be integrated within various model-based workflows for complex operational applications in water resources and beyond.

Table of Contents

List of Tables

List of Figures

Acknowledgements

First, I would like to acknowledge the financial support for my Ph.D. studies from research funding awarded to Professor Xu Liang, which includes award #NNX12AQ25G from the National Aeronautics and Space Administration (NASA) of the United States, award #OASRTRS-14-H-PIT from the United States Department of Transportation, and award Pitt #WO009 from the Pennsylvania Department of Transportation.

I would also like to acknowledge the assistance and contribution of Daniel Luna and Daniel Salas from the Department of Civil and Environmental Engineering at the University of Pittsburgh, and of Rui Wang, Sricharan Lochan, and Prof. Yao Liang from the Department of Computer & Information Science at the Indiana University-Purdue University Indianapolis for their contributions in the development of the Hydrologic Disaster Forecast and Response (HDFR) system. I would like to thank the Visiting Ph.D. student, Mr. Ruochen Sun, for his contributions in the implementation of Professor Liang's ideas on the development and testing of the HIP-POP algorithm.

Computational work related to this research was supported in part by the University of Pittsburgh Center for Research Computing through the provision of access to the SMP computer cluster.

1.0 Introduction

Hydrologic modeling is essential for the design of physical infrastructure, from roads and bridges to dams and channels; for planning day-to-day water use operations such as drinking water supply, irrigation, hydroelectric generation, ecological monitoring, navigation, etc.; and for providing early warnings and decision-support for dangerous extreme events such as floods, droughts, and landslides. Fortunately, the applicability and accuracy of hydrologic models has been greatly improved thanks to developments on three related threads.

The first one is the growing availability of *remote sensing* data products, which allow to monitor the evolution of key spatially-distributed environmental variables in near real-time. Noteworthy efforts include the deployment of Doppler radar networks to estimate high-resolution precipitation fields on several countries [1], and the fleet of Earth-monitoring satellites such as those from the MODIS [2], GPM [3], and SMAP [4] missions. Not only is all this information invaluable towards watershed modeling efforts, but it also enhances the predictive skill of meteorological forecast systems such as GFS [5], NAM [6], and ECMWF [7] whose atmospheric outputs are essential for hydrologic prediction.

The second thread is related to the development of *physically-based modeling* frameworks, in which the water and energy exchanges occurring on the land surface are simulated based on measurable quantities and using equations derived from fundamental laws of physics. As opposed to more traditional empirical or conceptual frameworks, simulation engines such as VIC [8], Noah [9], and DHSVM [10] can take advantage of detailed information on topography, land coverage, vegetation types, and soil characteristics. This explicit simulation of land-surface phenomena has extended the applicability of these methods beyond the prediction of discharge volumes at the output of watersheds: the accounting of variables such as soil moisture, evapotranspiration, snow cover, etc., allows for numerous additional applications such as those in collaboration with disciplines like meteorology [11]–[13], ecology [14], and geotechnics [15].

The third thread of developments that has significantly increased the potential of hydrologic modeling is the ever-increasing availability of *computational resources*. Computers are able to run models that accurately represent complex systems by discretizing the spatial and temporal domain into large numbers of individual interconnected elements. The aforementioned hydrologic modeling frameworks draw their power from the ability to solve the governing equations for each of a large number of modeling units and for arbitrarily-small time intervals. Moreover, in many cases, the performance of these numerical schemes can be hugely accelerated through parallelism by the use of multi-threaded CPUs, GPUs, and computer clusters [16]–[19].

While these developments offer countless opportunities to advance the reach of hydrologic modeling, they need to be adequately leveraged in order to unleash their full potential. Many challenges exist in this area and here we call the attention to two of them, which we consider of paramount importance. The first one is the bridging of the gap between all the hydro-meteorological data available and operational modeling. As identified before [20], even though many of these datasets are available for free on the internet, the heterogeneity of sources, transport protocols, formats, temporal and spatial resolutions, lag-times, accuracies, among others; constitute obstacles to the efficient incorporation of these information assets within models. Additionally, modelers typically need to resort to a series of diverse software tools, with varying degrees of ease of use, to acquire and process data; construct, calibrate, maintain, and run models; and visualize, analyze, and publish results.

The second grand challenge under consideration arises from the requirements of high-resolution sub-surface information in distributed physically-based modeling. The enhanced physical realism of modern modeling engines [21] is based on the ability to represent the heterogeneities of the land's characteristics in sufficient detail [22]. These high-resolution representations allow resolving the system's dynamics in a highly distributed manner, which translates into better accuracy. However, there is a steep price to pay in the introduction of multiple sources of uncertainty and of numerous unknowns, complicating our ability to reach adequate estimates. Here we classify the model uncertainty into three categories:

1. ***Structural uncertainty***: related to the choice of spatial and temporal resolution, inclusion/exclusion of modeling elements and phenomena (e.g., dams, agriculture, street sweeping), selection of governing equations, and element connectivity (e.g., drainage network topology).

2. ***Parameter uncertainty***: originating in the assignments to the model's parameters (e.g., soil porosity and conductivity, surface coverage type, etc.) and their spatial distribution in the watershed.

3. ***Initial state uncertainty***: corresponding to the selection of the initial conditions of the model; i.e., the assignment and distribution of state variables such as soil moisture and temperature.

In this dissertation we propose to address these two challenges through the development of an integrated hydrologic modeling system which unifies and streamlines the process of data acquisition; model building, deployment, and operation; and result visualization and analysis; and that incorporates state of the art optimization and machine learning techniques to allow for the sensible reduction of these main types of uncertainty, in order to take full advantage of distributed physically-based models.

The dissertation is organized in eight chapters, each of which addresses specific research questions. The various chapters expand on the motivation, goals, proposed approach, and current level of progress of its area of focus. Chapter 2 describes the development of the framework for the integrated modeling system. Chapter 3 and Chapter 4 introduce tools to address parameter uncertainty through the implementation of two calibration approaches. Chapter 5 introduces the problem of initial state estimation and a hybridized data assimilation algorithm to attempt to solve it. Chapter 6 and Chapter 7 expand upon the developed assimilation algorithm by introducing tools to more efficiently deal with initial state uncertainty in scenarios with complex models with numerous state variables. Finally, Chapter 8 recapitulates the main contributions of this dissertation and discusses how all the methods introduced can be integrated to offer researchers and practitioners a comprehensive framework for the creation of models that are useful for multiple hydrologic prediction applications.

2.0 Integrated Hydrologic Modeling Framework

This chapter introduces a software system whose objective is to ease complicated processes based on hydrologic modeling, through the integration of access to environmental information, standardized data exchange, model preprocessing, simulation engines, and result visualization tools. The chapter is based on the article “The HDFR System: Bridging the Gap between Environmental Sensing and Hydrologic Modeling” by Daniel Luna, Felipe Hernández, Yao Liang, and Xu Liang, currently in preparation.

2.1 Introduction

In the field of hydrologic modeling, there exists a major gap between the wealth of both environmental observations and modeling software, and the prompt accessibility to these tools by the broader community of potential end users. On one hand, the numerous data products from federal agencies do not have unified hosting sources, transfer protocols, data formats, and geographical projections. This results in modelers typically needing to resort to a series of diverse software tools, with varying degrees of ease of use, to acquire and transform data into useful forms [20].

On the other hand, there does not exist hydrologic modeling software which combines state-of-the-art capabilities, ease of use, and prompt accessibility. Basic hydrologic modeling tools, like the Army Corps of Engineers’ HEC-HMS [23], the Environmental Protection Agency’s SWMM [24], and the US Department of Agriculture’s WinTR-55 [25], provide modest capabilities and are available for free, making them very popular within the user community. Commercial packages, on the other hand, offer additional utilities; such as better modeling capabilities, more potent user interfaces, improved support, and/or enhanced interactivity with other software; at a cost to the users. Examples include Flood Modeller Pro [26], InfoSWMM [27], and the Watershed Modeling System [28].

On the other end of the spectrum, the best advances in representing geophysical phenomena can be found in scientific software. However, these packages require considerable efforts to be effectively used due to their experimental nature, their demands for theoretical expertise and input information, and their general lack of streamlined user interfaces. Such modern distributed hydrologic modeling engines include CLM4 [29], DHSVM [10], MIKE-SHE [30], Noah [31], TOPMODEL [32], and VIC [8].

The high entry cost for utilizing advanced data and modeling engines have turned the process of acquiring and formatting data, preparing and calibrating models, and performing simulations and sharing results, into a complex big-data challenge. As a response, researchers have proposed multiple systems that attempt to combine several of the “building blocks” that make part of these modeling workflows.

For example, the HydroDesktop [33] geographic information system (GIS) and HydroClient [34], its web counterpart, provide some level of access to remote data and preprocessing components. Similarly, the HydroShare platform [35], which started as a service for hosting and sharing hydrologic-related data with the community, has evolved through the introduction of multiple apps (like SWATShare [36]) into a more capable framework that allows for data acquisition, pre-processing, modeling, and visualization.

The orchestration of multiple components or modules can be achieved through software packages designed specifically for the integration of lower-level tools. Examples in the literature include Pipistrelle [37] (which integrates modules compliant with the OpenMI [38] standard), CSDMS' Web Modeling Tool (WMT) [39], [40] (which uses the BMI interface), and OMS [41] and ESMF [42], which allow the interconnection of multiple modeling and support components, and that count with a modest library of user-generated modules. However, while useful in executing complicated modeling workflows, the broad target disciplines targeted by these approaches, and the need for assembling a set of tools from the available collection, fail to provide modelers in hydrology with a streamlined and easily-accessible experience. Delft-FEWS [43], on the other hand, does combine data, modeling, and analysis components under a single user interface. Unfortunately, the package requires commercial licenses for operational deployment.

Given that the need for one-stop, high-quality hydrologic modeling system capable of remote data retrieval from diverse data sources remains unmet, here we introduce the Hydrologic Disaster Forecasting and Response (HDFR) system, an integrated framework that aims at bridging the large gap between advanced hydrologic modeling resources and the growing wealth of Earth-observing data for the community. Unlike the aforementioned systems in the literature, the HDFR is designed to single-handedly execute complex hydrologic workflows while simultaneously offering a capable and friendly user interface that will be open to the community at its due time. The HDFR features access to multiple hydrometeorological data sources, including NASA's GPM satellite-based precipitation mission [3] and SMAP satellite-based soil moisture mission [4], and to high-resolution distributed hydrologic modeling through the DHSVM [10], [44]. The remainder of this chapter is organized as follows: in Section 2.2 we describe the HDFR and its components; in Section 2.3 we demonstrate the system's capabilities with two example workflows; and in

Section 2.4 we offer a discussion and overview of the system together with the perspectives for its continuing development.

2.2 Proposed System

2.2.1 Overview

The Hydrologic Disaster Forecasting and Response (HDFR) system was developed as a series of extensions to the Geographic Resources Analysis Support System (GRASS) [45], an open source GIS which has numerous analysis, visualization, and management capabilities for spatial information. Despite the abundance of open source GIS packages available [46]–[49], GRASS provides special and unique features, which made it ideal for the development of the HDFR. For instance, its temporal framework opens the possibility of creating, visualizing, and managing spatiotemporal datasets, which strongly fit the nature of hydrologic and weather-related variables. Having been under continuous development since 1984, GRASS has cultivated an active community which is constantly contributing to its improvement. Additionally, this GIS offers multiple options for visualization, as well as a plethora of import/export formats to ease the communication with external software. Moreover, GRASS includes numerous modules for hydrologic model pre-processing and terrain analysis.

Many modules and basic operations in GRASS have large computational requirements and were thus developed in C++. However, its user interface and other high-level procedures, such as the temporal dataset framework, are written in Python through PyGRASS [50]—an application programming interface (API) where Python modules can access services from the C++ core set.

The HDFR system extends the module collection offered by GRASS by taking advantage of its extensible architecture and adding a new set of tools developed predominantly in Python, with some modules that still perform most computations through compiled C++ executables. The component architecture of the integrated GRASS-HDFR system is illustrated in Figure 1. The following subsections introduce the most relevant modules developed so far.

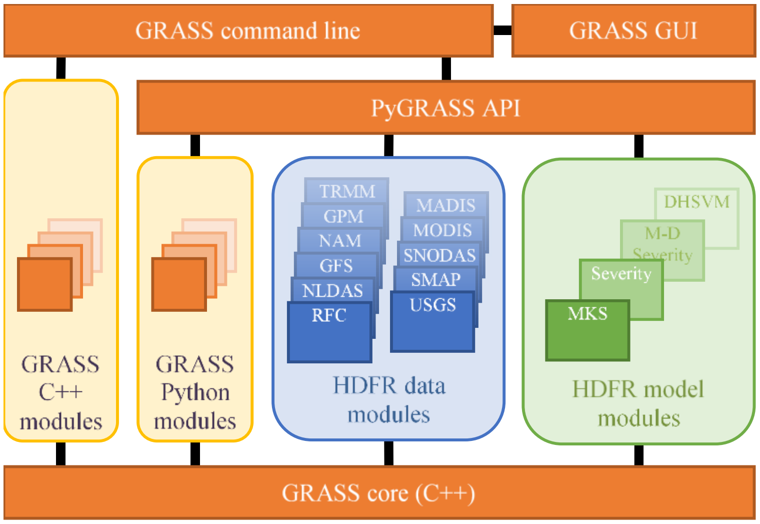


Figure 1 Component Architecture of the Integrated GRASS-HDFR system

Boxes indicate software components, while rounded rectangles indicate groups of modules. The HDFR modules are connected to the core system by interacting with the PyGRASS API.

2.2.2 Remote Data Retrieval

Table 1 summarizes all the data modules developed for the HDFR, together with their hosting agency, the variables available, how their data is obtained (e.g., through measurements or models), and their spatio-temporal extents and resolutions. Each data module is responsible for taking queries from the user with a desired spatio-temporal extent, contacting the remote server, downloading the files, and importing the information into the current GRASS session in one of the supported formats—typically time-evolving vector and raster types.

Table 1 List of Data Modules Available in the HDFR

Agency and Dataset	Variables; Source	Spatial extent, resolution	Temporal extent, resolution[a]
NWS' RFC precipitation [51]	Precipitation; multi-sensor (primarily NEXRAD [1])	Continental U.S., 4 km grid	Since 2013, hourly, 1 h lag,
NASA's NLDAS-2 [52]	Multiple meteorological; land stations	Continental U.S., 12 km grid	Since 1979, hourly, 4 d lag
NWS' GFS [5]	Multiple hydrometeorological; atmospheric/land-surface model	Global, ¼°-2.5° grids	3 hour, every 6 h, 192 h lead time
NWS' NAM [6]	Multiple hydrometeorological; atmospheric/land-surface model	North America, 12 km grid	Hourly, every 3 h, 60 h lead time
NASA's GPM [3]	Precipitation; satellite radiometer	Global, 0.1° grid	Since 2014, 30 min, 6 h lag
NASA's TRMM [53]	Precipitation; satellite radar	40°S – 40°N, 0.25° grid	2000 – 2017, 3 h, 6 h lag
USGS' water data [54]	Gage height, discharge; hydrometric stations	United States, > 10,000 sites	Per site
NASA's SMAP [4]	Soil moisture; satellite radiometer	Global, 40 km grid	Since 2015, daily, 50 h lag
NOAA's SNODAS [55]	Snow data; multi-sensor	United States, 0.7 km grid	Since 2003, daily, 4 h lag
NASA's MODIS snow [56]	Snow data; satellite	Global, 0.05° grid	Since 2000, daily, 3 h lag
NOAA's MADIS precip. [57]	Precipitation; rain gauges	Global	Per site

[a] Lag represents the amount of time between the actual measurement and the moment data is available to users. Note that some products, like GPM, offer estimates with different levels of quality control, each with a different lag time.

Special focus was placed on the development of precipitation data modules due to the key role of precipitation in driving hydrological processes and to the relative strengths and weaknesses of the multiple measurement techniques. While being the most accurate, point measurements, like those available through the Meteorological Assimilation Data Ingest System (MADIS) [57], offer the worst spatial coverage. The United States, as other developed countries, has filled this gap through the construction of a network of land-based Doppler precipitation radars [1]. The HDFR offers access to calibrated radar-based estimates from NOAA’s River Forecast Centers (RFC) [51]. For territories lacking such a network, distributed precipitation estimates can be obtained from instruments installed on space-borne platforms such as the now-defunct Tropical Rainfall Measuring Mission (TRMM) spacecraft [53], or the modern network of satellites that comprise the Global Precipitation Measurement (GPM) project [3].

Additional variables, like air temperature, air pressure, and air humidity, are also available through MADIS. Similarly, the North American Land Data Assimilation System (NLDAS) [52] offers multiple meteorological observations, but on a regular grid interpolated from point sensors. Forecasted variables from the U.S. National Weather Service for the atmosphere and the land surface can be accessed through the planet-scale Global Forecast System (GFS) [5] and the high-resolution North American Mesoscale Forecast System (NAM) [6] modules. Other variables available through the HDFR include river discharge from USGS stations [54], soil moisture from NASA’s Soil Moisture Active-Passive (SMAP) satellite [4], snow cover from NOAA’s Snow Data Assimilation System (SNODAS) [55], and snow cover from NASA’s Moderate-resolution Imaging Spectroradiometer (MODIS) satellites [56].

Table 2 shows the portals on which the information from all modules is hosted, the web protocol utilized for its acquisition, the file format that is obtained, and its geographical projection. Four of the HDFR’s modules were implemented by communicating with NASA’s Simple Subset Wizard (SSW), an online platform that hosts multiple datasets and offers spatio-temporal subsetting capabilities through an OPeNDAP [58] interface. Python’s urllib library was used to communicate with the remote servers in most modules, but some of them used executables developed in C++ that utilized the cURL library instead. Downloaded files are usually in purpose-specific binary formats like Network Common Data Form (NetCDF) [59], Gridded Binary

(GRIB), and Hierarchical Data Format (HDF). GRASS offers native support for NetCDF and HDF files, and NOAA's wgrib2 utility was utilized to interpret GRIB files.

Table 2 Access to Data on Remote Servers

Agency and Dataset	**Portal, *Protocol***	**File Format, *projection***
NWS' RFC precipitation	EDD, *HTTP*	GIS DBF, *HRAP* [1]
NASA's NLDAS-2	SSW, *OPeNDAP* [58]	Text, *Lat-lon*
NWS' GFS	Independent, *FTP*	GRIB2, *Lat-lon*
NWS' NAM	Independent , *FTP*	GRIB2, *Lambert*
NASA's GPM	SSW, *OPeNDAP*	NetCDF, *Lat-lon*
NASA's TRMM	SSW, *OPeNDAP*	NetCDF, *Lat-lon*
USGS' water data	Independent , *HTTP*	Text, *Lat-lon*
NASA's SMAP	SSW, *OPeNDAP* [58]	Text, *Lat-lon*
NOAA's SNODAS	Independent , *FTP*	Binary, *Lat-lon*
NASA's MODIS snow	Independent , *FTP*	HDF, *Lat-lon*
NOAA's MADIS precip.	Independent , *HTTP*	Text, *Lat-lon*

URLs of portals – EDD: http://preview.weather.gov/edd, SSW: https://ssw.gsfc.nasa.gov/, GFS: http://www.emc.ncep.noaa.gov/GFS, NAM: http://www.emc.ncep.noaa.gov/index.php?branch=NAM, USGS: https://waterdata.usgs.gov/nwis, SNODAS: http://nsidc.org/data/g02158, MODIS: https://modis.gsfc.nasa.gov/, MADIS: https://madis.ncep.noaa.gov/; *URLs of data servers* – RFC precipitation: http://www.srh.noaa.gov/data/ridge2/Precip/qpehourlyshape/, GFS: ftp://nomads.ncdc.noaa.gov/GFS/Grid4/, NAM: ftp://ftp.ncep.noaa.gov/pub/data/nccf/com/nam/prod/, USGS: https://waterdata.usgs.gov/nwis/dv?referred_module=sw&search_criteria=lat_long_bounding_box&submitted_form=introduction, SNODAS: ftp://sidads.colorado.edu/DATASETS/NOAA/G02158/, MODIS: ftp://n5eil01u.ecs.nsidc.org/SAN/MOSA/MYD10C1.005, ftp://n5eil01u.ecs.nsidc.org/SAN/MOST/MOD10C1.005, MADIS: https://madis-data.ncep.noaa.gov/public/sfcdumpguest.html

2.2.3 Modeling Capabilities

A series of model modules were developed within the HDFR to support scientific and engineering hydrologic applications that make use of the information available through the data modules. As opposed to data modules, model modules usually have more complex inputs, and their implementation usually involves a series of local computations instead of network communications. Here we describe some of the most relevant.

2.2.3.1 Data Fusion

In cases where multiple spatial observations of the same physical quantity are available through different sources, it is usually be desirable to “fuse” (combine) them to produce a single estimate [60]. This is of utmost utility when each of the data sources has their own strengths and weaknesses, for example in the case of rain gauge measurements (accurate but sparse) and satellite precipitation estimates (distributed but less accurate).

Data fusion is enabled in the HDFR through a module which implements a variant of the Multiscale Kalman Smoother (MKS) algorithm [61], [62]. The MKS discretizes the spatial resolution of gridded/raster information in a scale of integer powers of two (2^k), such that information can be easily transferred from one resolution to the next. After transforming every input to the 2^k representation, the algorithm performs an “upward sweep”to propagate the information up to the coarsest resolution, and then a “downward sweep” in the opposite direction. The error of each input is encoded through a variance value, which is then used in a Kalman gain matrix to allow computing weighted averages of the multiple estimates. The full description of the algorithm can be found in [61] and [63].

The MKS implementation in the HDFR includes a series of enhancements to lift the constraints of the original algorithm. First, it allows for multiple inputs at the same scale, and for inputs to have “missing data” values—which enables the use of incomplete data or the inputting of point information (e.g., from rain gauges) in the required raster form. Second, a wrapper was implemented to take inputs of arbitrary extent, resolution, and shape and transform them to fit the MKS’ representation of squared areas of identical extent within the 2^k hierarchy.

2.2.3.2 Event Severity Assessment

A second module allows assessing the severity of precipitation events through the computation of a series of maps that track the evolution of the return-period at each time step. The return period can be computed from a frequency analysis on historical precipitation [64] and is intended as an estimate of the average time frame that is expected between events of the observed magnitude. The HDFR currently uses frequency data derived from NOAA’s ATLAS 14 project [65]. Figure 2 shows an example of the input and output of this module.

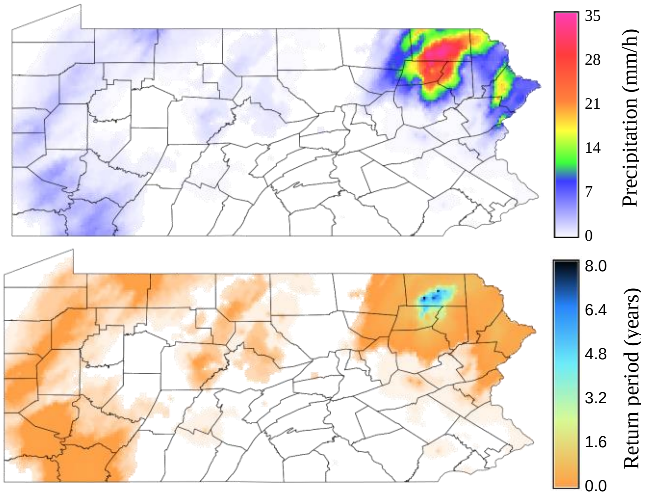


Figure 2 Example input and output of the precipitation severity module

Top: Input precipitation event on 09/30/2015 at 9am UTC over Pennsylvania. Bottom: Return period map of the same event as computed by the precipitation severity module.

A more thorough severity assessment of the precipitation is available through another module which performs temporal aggregations of the precipitation maps over time. Figure 3 illustrates how an event is aggregated in time windows of different user-defined durations. All possible instances are considered for each duration, producing new precipitation maps that are thereupon transformed into return-period maps. After this process, the maximum return period values for each pixel are extracted to produce a unique map with the maximum. As opposed to the previous module, this one thus allows to distinguish between fast events (as those associated with flash floods) and slow events (associated with alluvial floods).

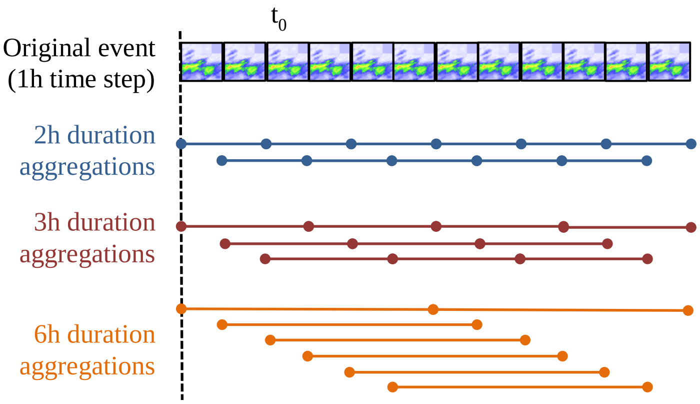


Figure 3 Temporal Aggregation of Precipitation for Severity Analysis

Aggregation time windows: 2 hours, 3 hours, and 6 hours. The aggregations are used to compute return periods for different types of storms (e.g., short and long).

2.2.3.3 Hydrologic Modeling

Finally, a module was developed to perform hydrologic simulations using the Distributed Hydrology Soil Vegetation Model (DHSVM) [10], [44]. The DHSVM can simulate subsurface flows using either kinematic or diffusion wave approximations, free-surface flow routing using conventional or kinematic wave approximations, and flows in channel networks using the Muskingum method. The three representations (subsurface, surface, and channels) are tightly coupled and support the distributed simulation of multiple phenomena (e.g. evapotranspiration, snow accumulation, soil moisture). The DHSVM is intended for spatial resolutions corresponding to those of digital elevation maps.

The preparation of the files for running the DHSVM is usually a very long manual process, which might be accelerated by the use of existing scripts but that still lack the benefits of a graphical interface. On the other hand, HDFR’s DHSVM module allows users to automatically create all of its required input files. From a digital elevation model, information on the vegetation cover and the soil, and the selection of an outlet/concentration point, the module can delineate the boundaries of the watershed, determine flow directions, delineate the channel network, assign tentative dimensions to each of the reaches, produce the necessary input files, and launch the modeling engine. Figure 4 shows an example of a set of inputs and outputs of the module for the West Branch Susquehanna River in Pennsylvania.

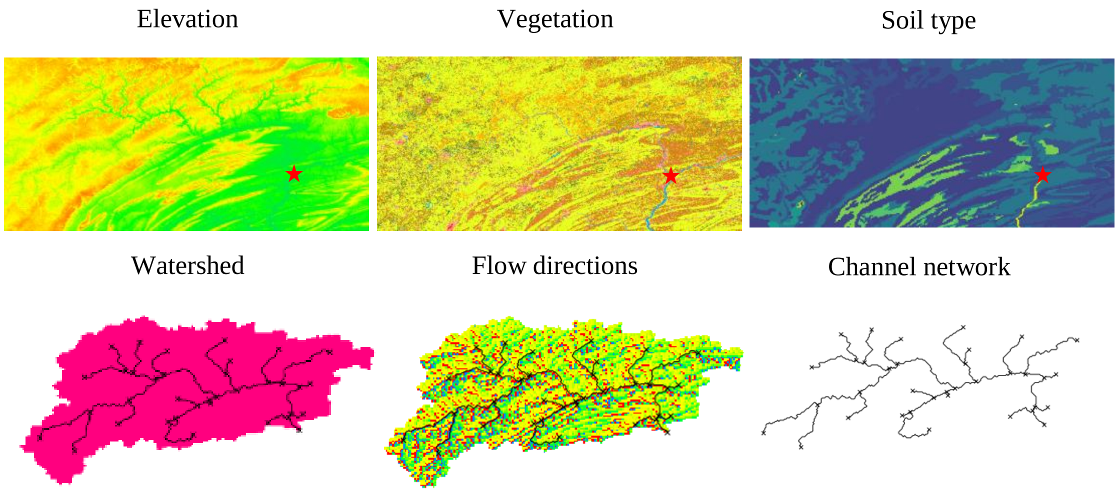


Figure 4 Inputs and Outputs of the DHSVM Module

Top (inputs): elevation, vegetation cover, soil type, and outlet (red star). Bottom (outputs): delineated watershed, flow directions, and channel network.

2.3 Capability Demonstration

In this section we demonstrate some of the capabilities of the HDFR through two examples. In the first one we retrieved a 6-hour storm over the state of Pennsylvania through the RFC precipitation module between 9/29/15 at 8 pm and 9/30/15 2 am UTC, as depicted in Figure 5. This precipitation event was used as the input for the multi-duration severity analysis module, which produced the maximum return period map, the map with the duration of maximum severity, and the map with the corresponding time stamp (the time at which the period of maximum severity begins) shown in Figure 6.

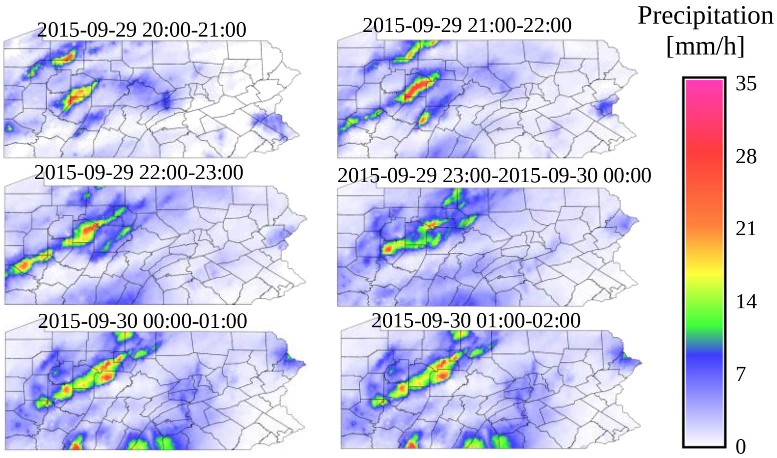


Figure 5 6-Hour Precipitation Event Over Pennsylvania

Obtained through the RFC precipitation data module.

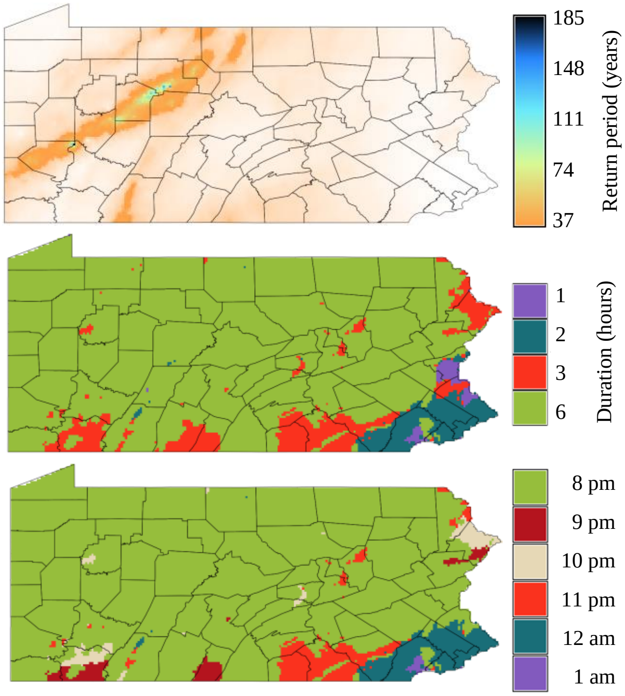


Figure 6 Example Output of the Multi-Duration Precipitation Severity Module

Output for the storm in Figure 5 (between 9/29 and 9/30, 2015). Top: maximum return period of precipitation. Middle: duration of the most severe period. Bottom: Start time of the most severe period.

The return period map was then used to assess the precipitation severity at the drainage area of a database of around 3000 scour-critical bridges monitored by the Pennsylvania Department of Transportation (PennDOT). Cross-referencing the vulnerability level of each bridge with the estimated return period within its drainage area, we were able to identify the list of bridges that required inspection due to the possibility of damage during the storm. Compared to PennDOT's current practice, the HDFR allows for the automatic retrieval of multiple precipitation products, their fusion, the use of meteorological forecasts for the analysis, and a much faster overall processing time.

In a second example, we created a hydrologic model of the Indiantown Run in Pennsylvania by inputting the required geographic information together with the coordinates of the desired outlet (76.599W, 40.441N). The DHSVM module produced outputs in the same fashion as those shown in Figure 4. Each of the channels is automatically assigned a cross-section geometry based on the watershed's average streamflow (selected by the user), their upstream drainage area, and their slope as extracted from the input elevation information. Outputs from the DHSVM include the streamflow at the outlet, which can be seen for this example in Figure 7.

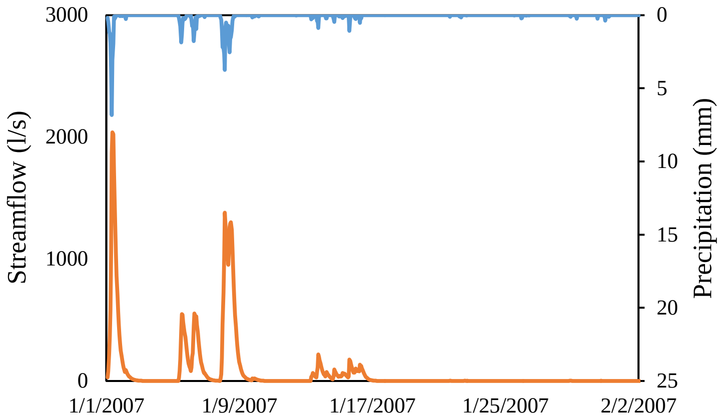


Figure 7 Example Rainfall-Runoff Simulation Results for the Indiantown Run Obtained using HDFR's DHSVM module.

2.4 Conclusions and Future Work

In this chapter we introduced the Hydrologic Disaster Forecasting and Response (HDFR) system, an extension to the open-source GRASS GIS, which was conceived and developed to act as a single-stop for hydrologic modelers to enable the access to numerous sources of hydrometeorological information, and utilize that information in a series of hydrologic analyses with multiple applications.

Available data sources include in-situ measurements from weather and hydrographic stations; remote sensing products from Doppler precipitation radars in the US, Earth-monitoring satellites that measure precipitation, soil moisture, and snow cover; and numerical weather prediction models from the U.S. National Weather Service. As demonstrated, modeling applications include the ability to fuse multiple sources of information into single estimates, establishing the multi-duration severity of precipitation events, and automatically preparing and performing high-resolution numerical simulations of the hydrologic cycle using the Distributed Hydrology Soil Vegetation Model (DHSVM).

Ongoing development efforts on the HDFR will enhance the capabilities of the existing data and model modules, and will increase the number of data products available through the system as well as the availability of modeling tools. We are especially interested on integrating the calibration and data assimilation tools, which will be introduced in subsequent chapters, into the system to improve the predictive capabilities of model modules and further reduce the need to rely on additional complex software. With this planned set of future developments, and the availability of the system for free on the internet, we expect that the HDFR will fill the needs of many researchers and engineers tackling hydrologic challenges, and offer them a state-of-the-art set of tools to leverage the benefits of this golden age of environmental big data and numerical modeling.

3.0 Model Parameter Estimation I: Multi-Objective Evolutionary Optimization

This chapter introduces the problem of uncertainty in model parameterizations, one of the three main sources of uncertainty described in Chapter 1, and proposes an optimization method to serve as the foundation for the development of complex hydrologic model calibration tools.

3.1 Motivation

Information related to the Earth's surface is available in sufficient quality for the creation of hydrologic models. For example, for the United States, elevation data is easily available at a resolution of around 10 m from the USGS's 3D Elevation Program [66], and land cover type data is available at a resolution of around 30 m from the USGS's National Land Cover Database (NLCD) [67]. On the other hand, thanks to the difficulty of probing the sub-surface, the information available on the characteristics of the soil is much poorer in comparison. The CONUS-SOIL database [68] contains multiple soil characteristics, such as composition percentage, bulk density, porosity, and permeability for different horizontal layers, but in a grid with a resolution of one kilometer.

Frequently, additional soil properties need to be inferred from the soil texture classification alone and thus several authors provide tables from where typical or average values can be taken [11], [69]. However, many of these parameters have higher variations within classes than between them [70], adding a very high degree of uncertainty into any model parameterization. Together with the high sensitivity of hydrologic phenomena to the nature of soils, the very coarse resolution and high uncertainty of the available information makes their adequate characterization in models a huge challenge.

The challenge is already complicated enough for lumped models, in which the entire watershed is aggregated in a single modeling unit that assumes homogeneous conditions. Typically, finding a right combination of parameters that make the model behave in the expected way (i.e., appropriately simulate measured variables such as streamflow, soil moisture [71], evapotranspiration [72], and snow cover [73]) requires extensive knowledge and the means to try many different possible combinations—a process called *calibration* [74]. However, the abandonment of parsimony by dividing the study area into smaller units with distinct characteristics, which can be justified in terms of the advantages of having distributed accounts of several variables, leads to additional problems as the number of unknowns grows.

In these more complex semi-distributed and distributed representations, interactions between sub-elements usually lead to the existence of multiple parameterizations that result in similar behavior at local scopes. This phenomenon is called *equifinality* and can be very problematic for two reasons [75]. First, if there exists a “true” parameterization, it is obscured by those that appear equivalent—a problem called *non-identifiability*. Second, and most importantly, a behavior deemed “appropriate” at a local scope (e.g., for a specific variable or time period) does not necessarily indicate that the model will behave appropriately at a different one. It is therefore common for over-complex models to behave well during a training phase, but then fail to adequately simulate reality when the conditions change from those under which the calibration was performed (this problem is called *overfitting* [76]).

As an effect of this, while theoretically every cell in a distributed model could have different parameter assignments to reflect the heterogeneity of the terrain, in practice modelers should refrain from optimizing more than a few parameters given the normal volume of available information. As a result, distributed models ultimately end up using only a semi-distributed abstraction of the soil characteristics as cells are given soil parameter assignments from only a handful of soil type categories. To begin to address these challenges, this chapter introduces an ensemble multi-objective evolutionary optimization algorithm to be used in complex hydrologic model calibration problems.

3.2 Method

An optimization algorithm was built by modifying a previously-developed single-objective algorithm with similar characteristics: Multi-Algorithm Ensemble for Several-Threads Robust Optimization (MAESTRO)[1]. It follows a recent and successful approach which consists of making use of an ensemble of cooperating metaheuristics [77], [78] in an attempt to overcome the problems associated with the "no free lunch" theorems [79], [80]—that is, the inability of a single optimization strategy to perform consistently well on problems of different nature. The algorithm is based on AMALGAM [81] and AMALGAM-SO [82], also featuring an adaptive technique for selecting how intensively to query each of the low-level optimizers in the ensemble but, with the ability to run evaluations on parallel and to support discrete, continuous, or mixed decision-variable vectors.

The algorithm features three low-level meta-heuristics within its ensemble. The first one is a traditional genetic algorithm [83]. The second is a hybrid approach which combines the

[1] Available at https://github.com/felherc/MAESTRO.

principles behind Ant Colony Optimization (ACO) and Metropolis-Hastings random walks, and was therefore named MetroACO. The ACO framework, originally targeted at graph problems [84], has been re-formulated to tackle combinatorial problems [85], [86] and later continuous global optimization problems [87]. In this last work, new candidate solutions are sampled from probability distributions defined by an archive of high-performing solutions. New values are sampled independently from univariate kernel density probability distributions in which the kernels are centered on the values of the solutions in the archive (kernels are probability distributions centered on data points used to create smoothed probability density fields from a series of samples [88], [89]). One the other hand, while not originally an optimization algorithm, the Metropolis-Hastings approach can be similarly used to create new candidate solutions [90], but from a joint probability distribution instead.

In MetroACO, solutions in the population are first assigned weights depending on the ranking of their domination front in order to create a weighted multi-variable kernel density probability distribution [89]. A fraction of the values for a new candidate solutions is then sampled from marginalized univariate distributions (ACO phase), and finally the rest of the values are sampled from a joint distribution conditioned on the originally sampled values (Metropolis phase). The conditional kernel density distribution is computed using an approach similar to that of Hyndman *et al.* [91]. A parameter is used to determine which fraction of the values to sample on each phase, with the extremes corresponding to pure ACO or Metropolis approaches. MetroACO currently supports S-class and D-class kernels only, and uses Silverman's rule [88] to compute the kernel bandwidths.

The second low-level optimization algorithm implemented is a global optimization version of the popular Gradient Descent (GD) method. GD approaches are used intensively for convex optimization due to their theoretical soundness and their computational efficiency. However, in global optimization these approaches would get stuck in local optima in the absence of any additional mechanism to guide them, and their reliance on the gradient of the fitness function makes their direct application on most problems unfeasible (as gradients cannot generally be computed in an analytical way). Moreover, multi-objective optimization poses additional challenges to the family of GD methods as there is no single gradient but a group of them—one for each objective.

The developed GD algorithm was designed to overcome these three limitations. First, a population-based approach is used so that the search is not conducted on a single point as in the work by Dai *et al.* [92], but on a series of vicinities defined by neighboring points. Second, gradients are computed empirically for each of these vicinities, similar to how it is done by Hewlett *et al.* [93]. Finally, new candidate solutions are generated in the estimated direction of the Pareto front which is defined by a linear combination of all the available gradient vectors. The algorithm can be briefly summarized in the following steps:

1. Normalize all the solutions so that the values for the decision and fitness variables have a mean of zero and a standard deviation of one.
2. Randomly select the bases for the vicinities from the population.
3. Compute the distance matrix in the decision variable space between the bases and all the solutions in the population.
4. Populate each vicinity with the closest linearly-independent solutions so that the steepest descent direction can be established (for n variables, it takes $n + 1$ non-collinear points to compute the direction of the steepest descent).
5. Compute the steepest descent gradients for each objective and compute the magnitude of their sum for each vicinity [93].

6. Assign the number of solutions to be generated by each vicinity so that the expected value is proportional to the magnitude of the sum of the gradient vectors.

7. Generate the candidate solutions for each vicinity in a direction computed as a random linear combination of the gradients, and using a kernel centered on the point given by the resulting direction and the step size.

8. Convert the candidate solutions back to the original un-normalized decision variable space.

3.3 Results and Discussion

As an initial test case we selected the Indiantown Run watershed in the Appalachian Mountains in central Pennsylvania for our tests. This 14.78 km^2 watershed is defined by the USGS river monitoring station 01572950 and corresponds to the US Hydrologic Unit 02050305 [94]. According to the National Land Cover Database (NLCD) [67], it is only 0.83% impervious and comprised 84.41% of forests, as shown in Figure 8. The elevation varies between 153 and 412 m according to data from the Shuttle Radar Topography Mission (STRM) [95], with an average slope of 14.5%. Two major soil types are identified in the 1-km resolution CONUS-SOIL dataset [68]: silt loam at the center of the valley (51%) and sandy loam at the mountain ridges (49%). The average streamflow at the outlet measured at the monitoring station is around 300 l/s for the record period; and there is an average precipitation of 1176 mm/year (550 l/s), an average evapotranspiration of 528 mm/year (250 l/s), and an average air temperature of 10.9°C according to NLDAS-2 [52].

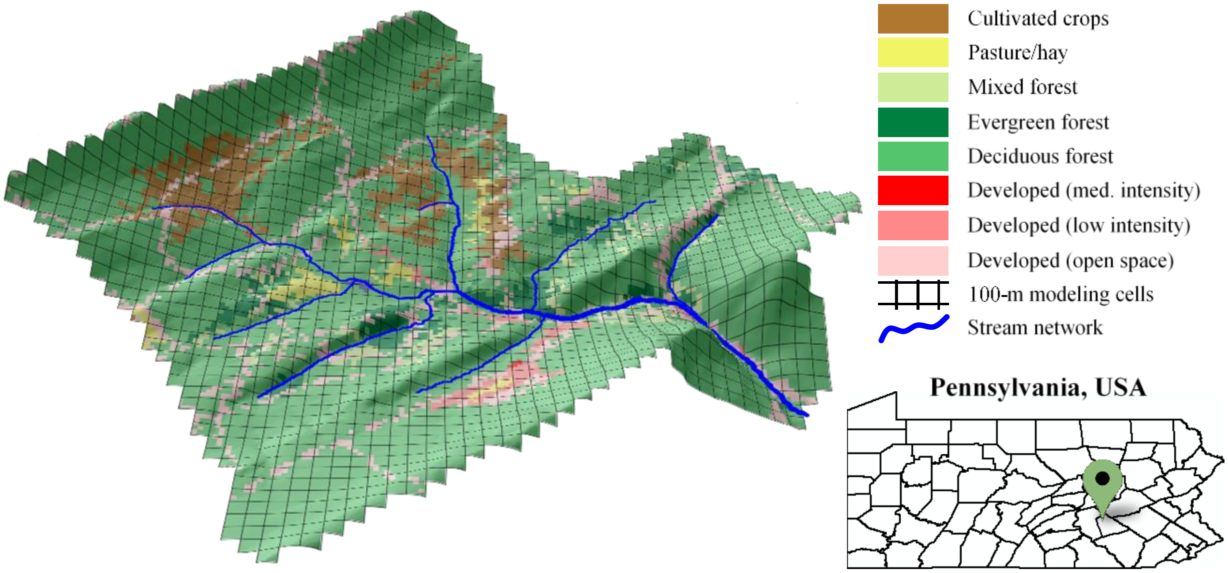


Figure 8 30-m Resolution Land Cover Distribution of the Indiantown Run Watershed

Land cover information obtained from the NLCD [67]**. The grid represents the cells of the 100-m resolution DHSVM model (1,472 in total), and the blue lines represent the channel network segments (21 in total) delineated based on elevation data from the STRM** [95]**.**

We created a 100-m resolution, 1-hour time-step model of the Indiantown Run watershed using the DHSVM with a total of 1,472 cells (which can be seen on Figure 8), four soil layers, and 21 stream segments. We first assigned the default parameters in the DHSVM documentation for each soil and land cover type. NLDAS-2 hourly forcing data [52] was then interpolated using Giovanni [96], and leaf-area index and albedo parameters were obtained from MODIS observations [2]. We then used fractional factorial experiments [97] to assess the effects of variations of multiple soil, vegetation, and stream parameters on the streamflow error, using hourly observations. This sensitivity analysis helped us determine a set of 18 parameters to be calibrated, including the conductivity, porosity, field capacity, and wilting point of the two soil types; the fractional cover and stomatal resistance of the deciduous forests; and the global Manning's coefficient of the streams.

We allowed for a 10-month spin-up run and then calibrated the model using hourly streamflow data between October, 2007 and June, 2009. Three optimization objectives guided the calibration: the traditional ℓ_2-norm Nash-Sutcliffe Efficiency (NSE_{ℓ_2}) coefficient [98], the mean absolute relative error (MARE), and the absolute bias. We chose these metrics from a larger set as we found them to be the least inter-correlated [74]. From a total of 2,575 candidate parameterizations, we selected the resulting parameters of one with a NSE_{ℓ_2} of 0.81, a MARE of 37.85%, and an absolute bias of 11.83 l/s. Figure 9 compares the hydrographs produced by the uncalibrated and calibrated models with the observed streamflow for 2008.

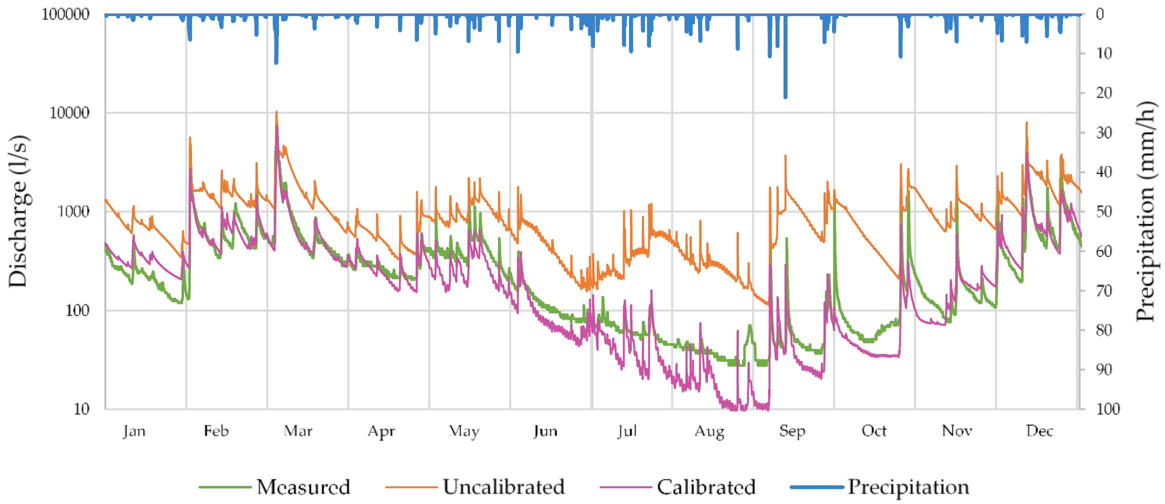


Figure 9. Comparison of Hydrographs for the Indiantown Run Watershed in 2008

As a second test case, we delineated a 10.44 ha catchment at the Beechwood Farms Nature Reserve in Pittsburgh. The catchment has been densely instrumented in a small sub-area with soil moisture sensors at different depths as part of a wireless sensor technology experiment [99]. Figure 10 shows an aerial view of the Beechwood Farms catchment together with the location of the soil moisture sensors and the 6525 4 m × 4 m cells of the developed model. The size of the circles indicate the number of available hourly observations available in the summer of 2016 (between April and September). The number of observations range from around 300 to around 2800, with interruptions occurring due mostly to network connectivity or battery power problems. The color of the circles correspond to the average soil moisture measured during that period at a depth of 30 cm. It can be seen that large differences in these means occur, even in adjacent areas with very similar conditions.

Statistical analyses involving the terrain characteristics at each point (slope, convergence, and flow accumulation) and adjacent vegetation from field surveys did not yield satisfactory relationships that could predict the observed variations in soil moisture, which lead to the conclusion that the sensor information has a very large degree of uncertainty. Two possible hypothesis that could explain this are the large levels of spatial heterogeneity of the soil conditions in the field and, more likely, variations in sensor and mote characteristics, installation procedures, and reading calibration methods. The second hypothesis has special validity because the sensor network has been installed and operated over several years with rotating personnel and unsteady supervision, and with heterogeneous technology.

Taking into account the inherent uncertainty of the data, we made a selection of 40 sensors (20 locations with readings at 10 cm and 30 cm depth) that had a relatively-large number of observations available, matched observations to adjacent sensors relatively well, portrayed soil moisture behavior that made physical sense, and that covered a wide-enough range of terrain and vegetation characteristics found in the area. A model was created to be run using the DHSVM with a temporal resolution of 15 minutes and a spatial resolution of 4 m (as shown in the grid in Figure 10). Both the meteorological forcings and the soil moisture observations were interpolated from their native resolution of one hour. Additionally, lacking any information on the spatial heterogeneity of soil characteristics in the catchment, we defined twelve sub-areas with roughly similar terrain and vegetation features to be assigned distinct parameter values on each during the calibration process.

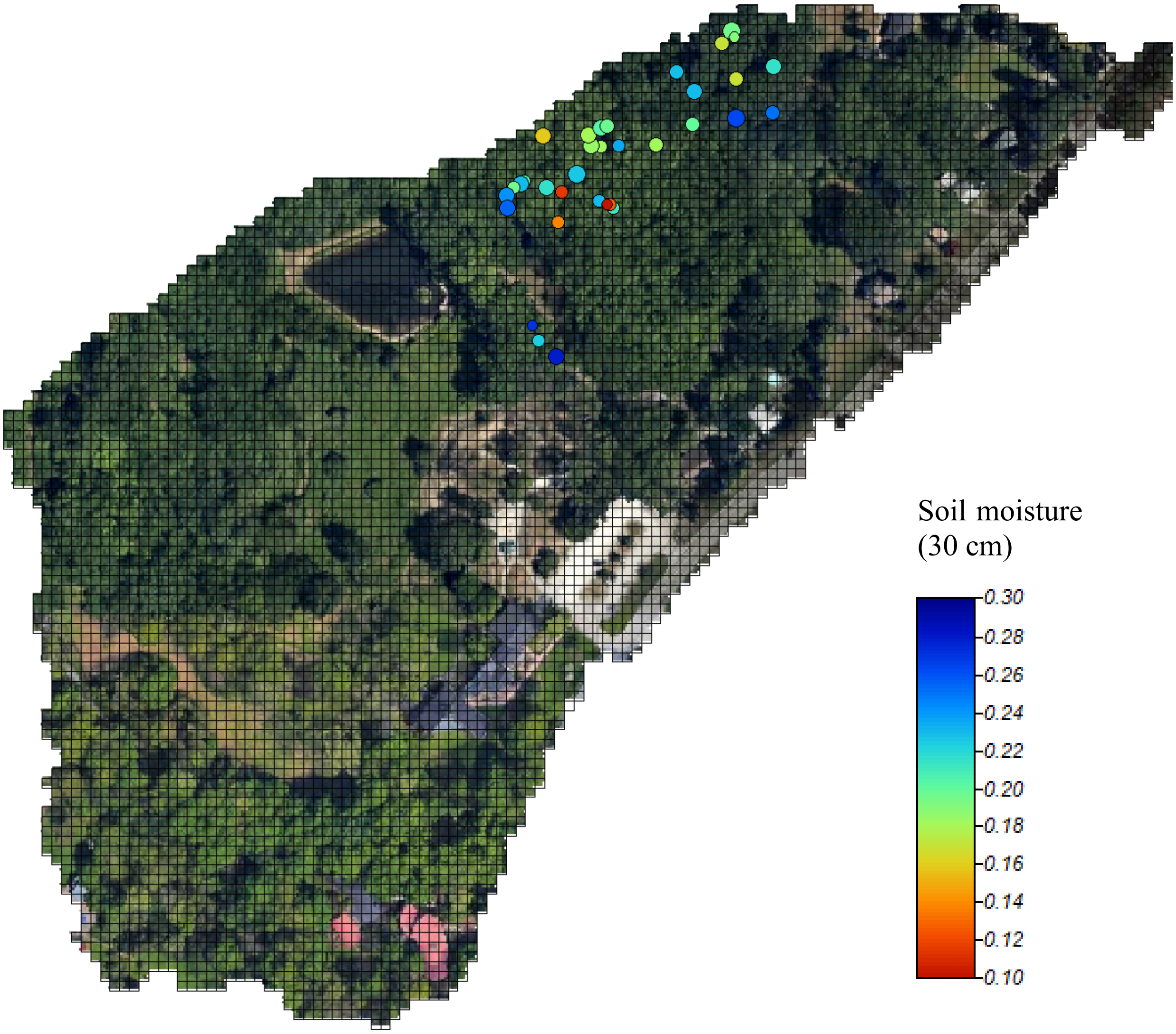


Figure 10 Aerial View of the Beechwood Farms Catchment

The grid represent the 4 m × 4 m cells in the DHSVM model. Circles indicate soil moisture sensor locations and their size correspond to the number of available hourly observations in the summer of 2016.

The model was calibrated using three optimization objectives: the average NSE_{l2} for all soil moisture time series at 10 cm depth, the average NSE_{l2} for all soil moisture time series at 30 cm depth, and the average NSE_{l2} from the best 20% of all the time series. The metrics were averaged by using the number of available observations at that location as a weighting factor. The selection of these three objectives stemmed from the desire to balance the overall performance of the model at different depths, the individual contributions of sensors, and the performance in cases where some of the sensors might have erroneous data (for the reasons mentioned above).

A total of 5332 candidate parameterizations were evaluated and, in the end, the resulting Pareto front had a total of 60 candidates. Figure 11 shows the observed and modeled soil moisture at three locations for one of those candidates with a good balance between the three objectives. The top panel shows a time series with a good fit; the middle panels shows an intermediate fit, where the model has a wet bias and a propensity to overestimate the peaks; and the lower panes shows a poor fit, in which the model displays a mostly homogeneous behavior interrupted by sudden peaks. In the last case, a closer analysis revealed that those sudden surges occur because of numerical instabilities that stem from the temporal resolution being too coarse for the spatial resolution for the case of the subsurface lateral routing. The vertical routing did not exhibit the same problem. Despite this, given that increasing the temporal resolution of the model would increase its computational requirements to the point where it becomes impractical for the planned applications, the calibrated model seems to be able to adequately simulate the intended phenomena to an adequate accuracy.

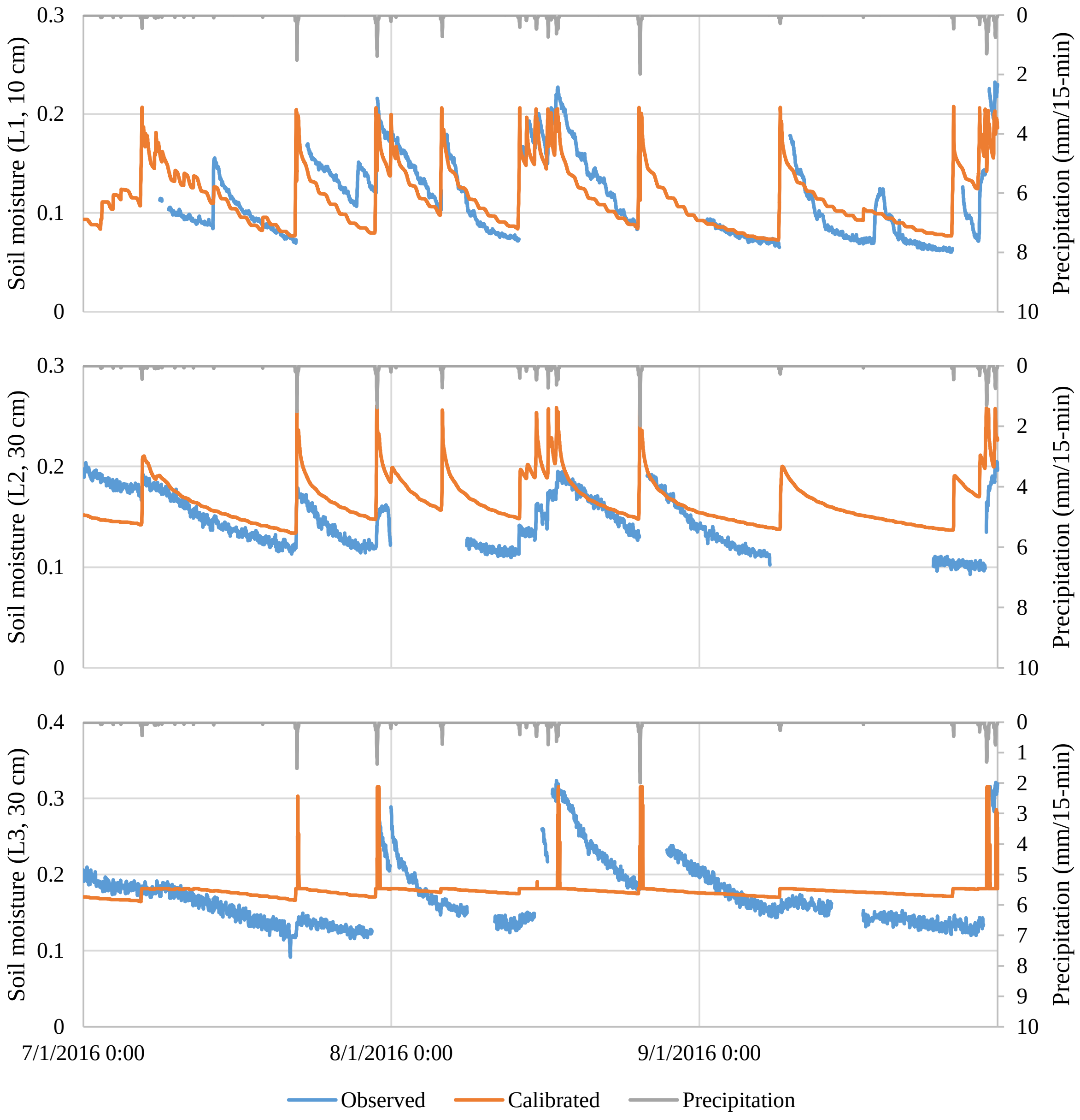


Figure 11 Observed and Modeled Soil Moisture at Three Locations

3.4 Conclusions and Future Work

In this chapter we developed a calibration algorithm for highly-distributed hydrologic models based on ensemble multi-objective evolutionary optimization. The algorithm combines the strengths of a genetic algorithm, a hybrid ant colony-Metropolis-Hastings optimizer, and a population-based gradient descent algorithm into a flexible ensemble that adapts to different types of problems and to the current conditions during the optimization process. The algorithm was coupled with the DHSVM modeling engine and was able to demonstrate good performance on two test watersheds in Pennsylvania.

Future work should focus on the incorporation of additional metaheuristic low-level optimizers into the algorithm, by attempting to enhance the dynamics between the strengths and weaknesses of the individual components. Moreover, additional techniques to adapt the ensemble to the current conditions can also be explored. For hydrologic calibration applications, existing methodologies should be improved to better manage the heterogeneous characteristics of natural soils. Moreover, many challenges remain related to the adequate selection and use of noisy and untrustworthy information and, more specifically, of distributed soil moisture observations.

4.0 Model Parameter Estimation II: Incorporating Expert Criteria

This chapter builds on the calibration algorithm introduced in Chapter 3 to create an enhanced method that incorporates expert criteria in a systematic way to improve the efficiency and physical soundness of uncertain model parameters and to mitigate the challenges posed by equifinality. The chapter is based on the article "Calibrating high-resolution hydrologic models reliably and fast: Optimization algorithms meet expert-knowledge principles" by Ruochen Sun, Felipe Hernández, Xu Liang, and Huiling Yuan. The article is currently in preparation to be submitted to the Water Resources Research journal.

4.1 Introduction

Despite the effectiveness of simple methods for aggregated hydrological estimation [100], [101], the development of complex prediction models [22] has enabled a wide array of applications, from large-scale monitoring [102], [103] to integrative frameworks spanning additional geosciences and engineering branches [13]–[15]. These tools have been made possible thanks to advances in the ability to simulate land-surface phenomena using physically-based approaches [21], together with the growing availability of powerful computational resources [18], [19]. Either because of their fine discretization of space and time [17], [104]–[106] or because of their massive scale [107], [108], these modern models are characterized by the immense amount of information that they process.

However, their increased complexity worsens the fact that almost all hydrologic models contain physical and/or conceptual parameters which cannot be measured directly, such as the porosity and hydraulic conductivity of the soils at different depths [109]. These sources of uncertainty have been addressed traditionally by adjusting the model's parameter values manually in such a way that the simulated response (typically streamflow) matches expected outcomes—usually in the form of available observations [110]–[113]. Applying expert knowledge to these manual calibration procedures has helped increase their accuracy and efficiency by following a set of principles: a) determination of realistic value ranges for the parameters, b) prioritization of those that are the most impactful and meaningful, c) establishing of cause-and-effect relationships between specific parameters and specific components of the response, d) addressing the problem sequentially by modifying only one or a few parameters at a time, and e) gradually narrowing the range of each estimate within a cycle of continuous improvement. Mastering the application of these principles allows for a relatively efficient search of adequate parameterizations in the rapidly-growing solution space when multiple unknown interacting parameters are involved, but requires a deep understanding of the physical phenomena behind the model and also lengthy practical training.

Despite attempts to systematize such manual calibration efforts [114], [115], the advent of more complex models necessitates the use of automated, time-efficient methods given the sheer magnitude of the solution space [116]. For instance, there is a whole body of literature dedicated to sensitivity analyses to determine parameter prioritization [117], [118]. But most significantly, there have been mayor advances in calibration algorithms based on breakthroughs in the field of optimization. Early methods, such as derivative-based algorithms [119] or direct search methods—like the Nelder-Mead simplex approach [120], utilized local optimization techniques to find locally-optimal solutions. Later studies have focused on advanced global optimization methods, which include genetic algorithms [83], [121], [122]; differential evolution [123]; particle swarm optimization [124], [125]; and shuffled complex evolution [126], [127]. In essence, these approaches take advantage of being able to run the model numerous times and use computational heuristics (in place of expert knowledge) to accelerate the search process.

While relatively successful, optimization methods come with their own set of limitations. Among them is the fact that, when a model possesses many interacting parameters, multiple combinations of value assignments might lead to similar responses—a phenomenon called equifinality [75], [128], [129]. An important consequence is that a parameterization obtained through direct optimization might not only be "right for the wrong reasons" [130]–[132]—and therefore potentially ill-suited for prediction purposes—but it will necessarily ignore the inherent uncertainty in parameter identifiability. Probabilistic calibration approaches were introduced to mitigate part of this problem by allowing the creation of stochastic estimates. They include the generalized likelihood uncertainty estimation (GLUE) method [133], [134], and a wealth of techniques based on Markov Chain Monte Carlo and Bayesian inference [135]–[142]. In second place, the dozens or hundreds of unknown parameters in complex distributed models still constitute a major challenge for the most advanced algorithms, and therefore the models require massive computational resources to be properly calibrated [125], [143], [144]. To address this specific problem, some researchers propose further attempting to improve the efficiency of search heuristics by, for example, limiting the number of parameters to modify at the same time [145], [146].

In this spirit, a common thread in the literature suggests that the problems stemming from high-dimensionality could be better tackled by incorporating more of the guiding principles of manual calibration into optimization-based methods, especially given the success of the resulting hybrids in other fields [147], [148] or given cases where expert algorithms have even managed to outperform the optimizers [149]. For example, cause-and-effect relationships between parameters and different parts of the resulting hydrographs [150], [151] inspired multi-objective calibration approaches [152], [153]—which have gathered considered attention [74]. Similarly, the iterative adjustment of parameter value ranges has been a major point of interest [141], [142], [144].

Therefore, in this article we present a hybrid calibration approach, named HIerarchical Prioritization for Parameter OPtimization (HIP-POP), which attempts to incorporate the expert-knowledge principles of manual methods into an optimization-based framework. HIP-POP simultaneously integrates all the strategies that have been proven so far in isolation, namely, reducing the size of the search space, establishing cause-and-effect relationships, and refining parameter ranges progressively. Additionally, it implements the principle of prioritizing the estimation of the most significant parameters, a strategy that helps in reasonably reducing uncertainty where it matters the most, attaining more realistic estimates, and increasing the overall computational efficiency of the entire process. The algorithm is described in detail in Section 2, the setup of a testing case study is explained in Section 3, and the results comparing our method with a traditional automatic approach are shown illustrated in Section 4. Conclusions are provided in Section 5.

4.2 Methodology

Let a hydrologic model be represented by a function $\boldsymbol{\mathcal{M}}$ which produces a vector of outputs $\boldsymbol{o}^{t+1}$ and a vector of state variables $\boldsymbol{s}^{t+1}$ at time $t+1$ from a vector of state variables $\boldsymbol{s}^{t}$ at time t, a vector of forcing values $\boldsymbol{f}^{t+1}$ at time $t+1$, and a vector of static parameters $\boldsymbol{p}$:

$$\boldsymbol{o}^{t+1}, \boldsymbol{s}^{t+1} = \boldsymbol{\mathcal{M}}(\boldsymbol{s}^{t}, \boldsymbol{f}^{t+1}, \boldsymbol{p}) \tag{1}$$

The outputs $\boldsymbol{o}$ correspond to fluxes such as the streamflow at each of the channels and the evapotranspiration from each sub-watershed (or cell/pixel if the model is gridded). The state variables $\boldsymbol{s}$ include the soil moisture and temperature, and the depth of the snow pack. Forcings $\boldsymbol{f}$ include air temperature, air pressure, and precipitation. The parameters $\boldsymbol{p}$ include the soil hydraulic conductivity and porosity, the percentage of impervious land, and the friction factor of the channels. The model can also be run as over an extended period between an initial time t_i and a final time t_f by iteratively using Eq. (6):

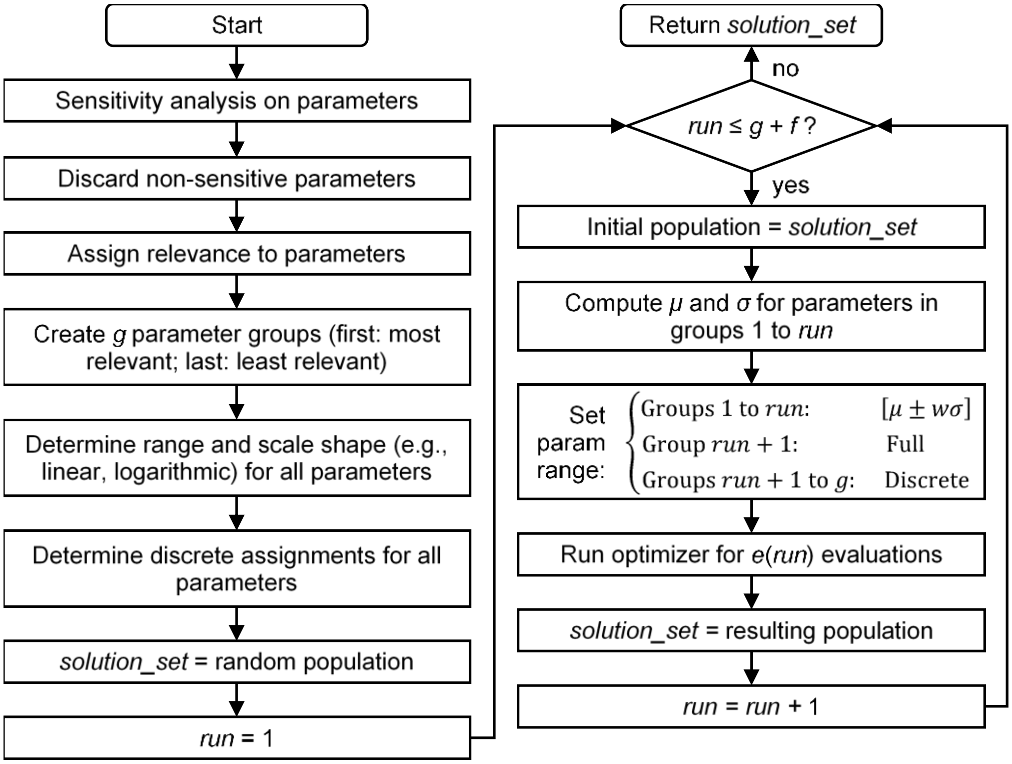
$$\boldsymbol{o}^{t_{i+1}:t_f} = \boldsymbol{\mathcal{M}}(\boldsymbol{s}^{t_i}, \boldsymbol{f}^{t_{i+1}:t_f}, \boldsymbol{p}) \tag{2}$$

For the results of the model $\boldsymbol{o}^{t_{i+1}:t_f}$ to be agreeable with reality, $\boldsymbol{s}^{t_i}$, $\boldsymbol{f}^{t_{i+1}:t_f}$, and $\boldsymbol{p}$ need to be properly estimated. Calibrating model M can be defined as the process of estimating the parameters $\boldsymbol{p}_*$ such that the model's output mimics a set of observations $\boldsymbol{o}_{\text{obs}}^{t_{i+1}:t_f}$ as much as possible, given known values for $\boldsymbol{s}^{t_i}$ and $\boldsymbol{f}^{t_{i+1}:t_f}$. That is, the calibration seeks to find $\boldsymbol{p}_*$ such that one or more error metrics $\boldsymbol{e}\left(\boldsymbol{o}^{t_{i+1}:t_f}, \boldsymbol{o}_{\text{obs}}^{t_{i+1}:t_f}\right)$ are minimized (note that if more than one error metric is used, $\boldsymbol{p}_*$ is not necessarily unique):

$$\boldsymbol{p}_* = \underset{\boldsymbol{p}}{\operatorname{argmin}}\, \boldsymbol{e}\left[M(\boldsymbol{s}^{t_i}, \boldsymbol{f}^{t_{i+1}:t_f}, \boldsymbol{p}), \boldsymbol{o}_{\text{obs}}^{t_{i+1}:t_f}\right] \tag{3}$$

The HIP-POP calibration algorithm, aimed at solving the problem posed in Eq. (3) for vectors $\boldsymbol{p}$ of high dimensionality, is divided into two phases: a strategy phase and an iterative optimization phase. The strategy phase serves as the planning stage for the optimization phase. It encompasses determining reasonable variation ranges for each of the parameters, establishing which of them are the most important, deciding in what order they should be estimated, and selecting the number and length of the iterations to be performed. This phase involves a modest amount of expertise from the user, but could be streamlined with the proper guidelines. On the other hand, the second phase can be fully automated and consists of running a population-based optimization algorithm iteratively. In each iteration, the focus is set on a different subset of the parameters, progressively reducing the uncertainty of the entire vector of estimates for $\boldsymbol{p}$.

Figure 12 shows the flow chart of HIP-POP, with the first phase consisting of the steps on the left and the second phase consisting of the steps on the right. The algorithm has four main meta-parameters: the number of parameter groups g, the multiplier of the standard deviation to establish the updated variation range of parameters w, the number of extra runs for the last grouping x, and a function to compute the number of solutions to evaluate given the run number $n(r)$. Here we detail each of the steps of the algorithm and the purpose of these meta-parameters.



Start
Sensitivity analysis on parameters
Discard non-sensitive parameters
Assign relevance to parameters
Create g parameter groups (first: most relevant; last: least relevant)
Determine range and scale shape (e.g., linear, logarithmic) for all parameters
Determine discrete assignments for all parameters
solution_set = random population
run = 1
Return *solution_set*
no
$run \leq g + f$?
yes
Initial population = *solution_set*
Compute μ and σ for parameters in groups 1 to *run*
Set param range:
Groups 1 to run: $[\mu \pm w\sigma]$
Group $run + 1$: Full
Groups $run + 1$ to g: Discrete
Run optimizer for $e(run)$ evaluations
solution_set = resulting population
run = *run* + 1

Figure 12 Flow Chart of the HIP-POP Approach

4.2.1 The Strategy Phase

The first four steps seek to identify the relative importance of each of the parameters of the model. Some might be discarded as unimportant right away based on experience, and will not be subjected to calibration; that is, these parameters are assigned default values and are left constant every time Eq. (2) is evaluated. For those that are not discarded in this way, a sensitivity analysis is performed in which different assignments for each parameter are tested to assess the effect on the error metrics $\boldsymbol{e}$ after running the model (Eq. (2)). The selection of these value assignments is typically done by establishing reasonable maxima and minima for the expected range of variation of the parameters (which requires expert knowledge). There are plenty of sensitivity analysis methods available to choose from in the literature, with varying levels of complexity, accuracy, and computational efficiency [117], [118].

Independently of the sensitivity method used, HIP-POP requires that the analysis provides enough information so that the user can organize sensitive parameters in groups of decreasing importance or relevance. That is, once the sensitivity analysis is performed, the user needs to sort the sensitive parameters into g groups, with the first group containing the most relevant parameters, and the last group containing the least relevant. The parameters in the first groups will be explored more thoroughly than those in the latter groups and, additionally, assignments to parameters in the latter groups will be done based on assignments to parameters in the first ones. That is, priority will be given to the most significant parameters while the rest need to accommodate accordingly. This stands in contrast to traditional optimization-based calibration methods in which all parameters are explored simultaneously, leading to a search in an immense solution space.

All the parameters that are found to be non-sensitive can be left constant with default value assignments. In selecting the number of parameter groups g we offer the following rules of thumb. If the second phase is not fully automated, larger values of g will result in additional processing steps in between runs of the optimization algorithm. On the other hand, if this is not a concern, increasing g results in a more detailed analysis that can emphasize the benefits of HIP-POP. However, we do not recommend separating equally-relevant parameters into different groups for the sake of increasing g, as this may translate into some of them being disproportionally de-emphasized.

The next step consists of determining the range of variation $\left[p_{i,\min}, p_{i,\max}\right]$ of each of the sensitive parameters p_i to be allowed within the search. Again, this range should be bounded by realistic values and, therefore, requires knowledge of the (physical) meaning of the parameters, which can be acquired from previous experience—e.g., existing tables can be utilized for standard soil textures. Additionally, using an exponential transformation to the variation range of specific parameters can be advantageous in some cases (e.g., the permeability of soils is spread more uniformly in the conductivity range when using an exponential scale).

In addition to the full allowed range of variation defined in the previous step, HIP-POP utilizes one or more discrete sets of values to be assigned to each parameter. For example, if one parameter is allowed to vary between 0 and 1, HIP-POP can also use the set $\{0.2, 0.5, 0.8\}$ as possible assignments to that parameter. When a group of parameters is being targeted specifically, the full range of variation is utilized. However, if another group is being targeted, using the much-constrained discrete set still allows for some degree of variation but does not exponentially increase the dimensionality of the search space. The determination of these discrete variation ranges will help in the optimization phase to control how much focus to give to specific parameter groups at different points in time.

Once the sensitive parameters have been assigned to g groups, and their full and discrete ranges of variation have been determined, the running plan to be used in the second phase, i.e., the optimization phase, can be established. Each "run" in HIP-POP corresponds to one invocation of the optimization algorithm, in which a subset of all the parameters are adjusted within their corresponding ranges to attempt to minimize the objective functions—the error metrics $\boldsymbol{e}$. HIP-POP uses multiple runs because the focus is shifted from one group of parameters to another in an attempt to simultaneously prioritize the most relevant ones and to accelerate the whole process by reducing the size of the search space in each run. The running plan consists of determining how many runs will there be in total, which groups of parameters to focus on in each run, and how many candidate solutions/parameterizations to be evaluated in each run (how many times to evaluate Eq. (2)).

4.2.2 The Optimization Phase

As a default option, Figure 12 offers a specific type of running plan in which only the first group is fully explored in the first run, the second group is fully explored in the second one, and so on. Moreover, less-sensitive parameters are allowed to vary according to their discrete value ranges in the first runs. This is done to avoid conditioning the assignments to the parameters in the first groups on single assignments to the parameters in the later groups. Additionally, once the run that focuses on a group is completed, parameters in that group are still allowed to vary in subsequent runs—but within a range of reduced size. This type of running plan is better depicted by the example in Table 3, in which, for example, in run two the parameters in the first group are constrained to a reduced range, those in the second group are allowed to vary within their full range, and all those in other groups use discrete variation ranges. The example optimization plan in Table 3 also includes an additional number $x = 1$ of runs at the end which explore all of the parameters simultaneously. Users are free to experiment with their own types of running plans, but we found that this configuration faithfully follows some of the guiding principles behind the design of HIP-POP: a) prioritization of sensitive parameters, b) a divide-and-conquer approach to constraint the size of the search problem, and c) a progressive reduction in the uncertainty in the estimation of the parameters.

Table 3 Setup of the Calibration Schemes Based on HIP-POP

	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7
Number of model running times							
	200	300	450	550	700	800	1000
Evolution of parameter ranges[a] *and number of parameter discrete values*							
Group1	full	shrunk	shrunk	shrunk	shrunk	shrunk	shrunk
Group2	5	full	shrunk	shrunk	shrunk	shrunk	shrunk
Group3	5	5	full	shrunk	shrunk	shrunk	shrunk
Group4	5	5	5	full	shrunk	shrunk	shrunk
Group5	5	5	5	5	full	shrunk	shrunk
Group6	2	2	2	3	5	full	shrunk

[a] As described in section 2, "full" means parameters are calibrated with original full, while "shrunk" means parameters are calibrated with updated range that has been shrunk after the last run.

The second phase uses a set of solutions or candidate parameterizations (assignments to $\boldsymbol{p}$) that evolves from run to run. In terms that are utilized in the field of evolutionary computation, this set is called a population (given that the conceptual origins of genetic algorithms stem from Biology). The set is initialized randomly by taking into account the parameter ranges established for the first run in the preparation strategy phase. If the plan dictates the use of the full range, values are sampled from a uniform distribution; if it dictates that a discrete range is used, only values from that range are randomly selected. After the first run, if a parameter has already been fully explored, then a shrunk variation range can be used for subsequent runs. From the population, the mean μ and standard deviation σ of the assignments for each parameter can be determined. The shrunk range for parameter p_i can be determined using meta-parameter w like this:

$$\max\left(p_{i,\min}, \mu_i - w\sigma_i\right) \leq p_i \leq \min\left(p_{i,\max}, \mu_i + w\sigma_i\right) \tag{4}$$

Large values of w allow for a more gradual convergence of the parameter estimates from one run to the next. Once the optimization algorithm has been configured with the adequate allowed ranges of variation for the parameters, the current population (or part of it) is set as the initial population of that particular optimization run (see Table 3). That is, aside from the first run, the optimization algorithm does not start from a completely random initial population, but from one that has already achieved a certain level of convergence towards the objective functions (from previous runs). However, allowing part of the run's initial population to be determined randomly should help in balancing the inherent conflict between exploitation and exploration [154] by increasing the solution diversity [155].

With its initial population defined, the optimization algorithm is allowed to run for a number of evaluations given by the function $n(r)$. Each evaluation involves computing the objective functions $\boldsymbol{e}$ for a given set of parameter values $\boldsymbol{p}$, which requires running the model (Eq. (2)) and comparing its output to the observations. The function $n(r)$, where r is the index of the current run, can be expressed as an actual function or in tabular form, and it is up to the user to determine it. In the example shown in Table 3, runs have progressively larger number of target evaluations. The reason behind this distribution is to allow for further exploration in the runs that have to deal with larger numbers of parameter groups. However, as before, users are encouraged to experiment with optimization plans that might better fit the needs of their specific applications.

Finally, once the optimization algorithm is done, the resulting population is set as the current population to be used in future runs or to be returned to the user as the final result (if no more runs have been scheduled). In the case of a single objective/error metric e, the population is made of the parameterizations/solutions that performed best in light of the objective. If multiple objectives/metrics $\boldsymbol{e}$ are used, this population may take the form of a Pareto Front [156], which includes solutions that performed well on each of the individual metrics and also those that offer the best trade-offs between them.

4.2.3 Optimization Algorithm

HIP-POP can be utilized with any optimization algorithm that is population based; i.e., one that is able to receive an initial population of solutions (in this case vectors of assignments to the parameters $\boldsymbol{p}$) and, similarly, return an optimized population that solves the problem stated in Eq. 3. If multiple error metrics $\boldsymbol{e}$ are defined, the optimizer needs to support multiple objectives or, alternatively, the various metrics need to be lumped into a single formula. In the literature there exist many evolutionary multi-objective algorithms that could be used [156]. However, in order to simultaneously provide continuous and discrete parameter variation ranges to multiple groups of parameters, the optimizer needs to be able to encode candidate solutions using both continuous and discrete variables. This last requirement is rare for optimization problems and, thus, there are not many such algorithms available—usually optimizers focus on “global” optimization problems (for continuous variables) or on “combinatorial” optimization problems (for discrete variables) [157]. If only the former are available, which are the ones generally used for geophysical model calibration, continuous encodings can be quantized to generate assignment for the discrete parameters. That is, continuous values produced by the algorithm need to be discretized by dividing the range of variation into a number of ranges that correspond to the discrete value assignments, and then assigning the corresponding discrete value.

In our implementation of HIP-POP we used an optimization algorithm which allows one to represent candidate solutions using both types of encodings simultaneously. This algorithm follows a recent and successful approach which consists of making use of an ensemble of cooperating metaheuristics [77], [78]. The simultaneous use of multiple low-level search heuristics allows to mitigate the problems associated with the “no free lunch” theorems [79], [80]: the inability of a single optimization strategy to perform consistently well on problems of different nature. The algorithm is similar to AMALGAM [81], [82], featuring an adaptive technique to determine how intensively to query each of the low-level optimizers in the ensemble and the ability to run evaluations in parallel. Two low-level metaheuristics were used within the ensemble: the established NSGA-II [83] and a hybrid between a Metropolis-Hastings sampler [90] and Ant Colony Optimization for discrete [158] and continuous [87] variables.

4.3 Experimental Design

In this section we first present how we designed a complex high-dimensional synthetic calibration problem to compare the performance of HIP-POP with that of a traditional evolutionary optimization-based calibration approach. Then we introduce three different configurations of HIP-POP and, finally, we define the performance metrics that were used to evaluate them.

4.3.1 Synthetic Case Study

While using real observations provides a more realistic benchmark to test the methodology with, measurements of soil characteristics are costly to obtain and only provide localized information. Therefore, we opted for a common approach in which the results are benchmarked against a known synthetic model [159], [160]. This reference model is used to generate a time series of outputs to calibrate the initial model against, and then the obtained parameters are compared to the reference values. However, in order to take into account the structural errors of models, which are prevalent when simulating natural processes, the reference model and the model to calibrate were both run using substantially different modeling engines. This makes our synthetic testing approach significantly different from others used in previous studies where the reference model and the model to calibrate are the same. By using two very different models we can test whether HIP-POP has the capability to overcome the model structure uncertainty and converge to more realistic parameter values than the traditional calibration approach.

We used two popular distributed hydrologic modeling engines, namely the Variable Infiltration Capacity (VIC) [8], [161]–[163] and the Distributed Hydrology Soil and Vegetation Model (DHSVM) [10], [44] to construct the synthetic experiment. VIC was originally designed to model large watersheds by taking into account of the effects of spatial sub-grid variability of precipitation, soil properties, and vegetation cover on soil moisture and surface fluxes (e.g., evapotranspiration). The DHSVM, on the other hand, was developed to numerically represent the effects of local weather, topography, soil type, and vegetation on hydrologic processes within relatively smaller watersheds with high spatial resolution. We applied the two distributed modeling engines to the French Creek watershed (Figure 13) in Pennsylvania. This watershed has a drainage area of about 160 km^2, which is suitable for both models. VIC was configured at a 1/32° resolution with a daily time step, while the DHSVM ran at a 500-m resolution with an hourly time step. The layout of the VIC modeling grid for this watershed is also shown in Figure 13. The daily discharge records of the basin outlet from 2003-2011 were collected from U.S. Geological Survey (USGS) station 01472157. When contrasting the simulated discharge, the time series produced by the DHSVM was aggregated to daily amounts. The meteorological forcing data used to run the two models was collected from Phase 2 of the North American Land Data Assimilation System (NLDAS-2) forcing dataset [52].

In this study we used VIC to generate the synthetic daily "observed" streamflow $\boldsymbol{o}_{\mathrm{obs}}^{t_{i+1}:t_f}$ from 2003 to 2011, and then we calibrated the DHSVM model against the synthetic streamflow by maximizing the Nash–Sutcliffe Efficiency coefficient (NSE) [98]. The calibrated parameters that are common between VIC and the DHSVM are the soil's field capacity, porosity, wilting point, vertical saturated hydraulic conductivity; and the vegetation's minimum stomatal resistance and radiation attenuation. Each of these parameters has an actual physical meaning and plays the same role in the two models. Therefore, the values of these parameters defined in the VIC model are used as reference "true" values for evaluating the calibration quality of the DHSVM model. In addition to these physically-based parameters, there are six more conceptual parameters (Table 4) in VIC which have significant impacts on simulated streamflow. Therefore, for the sake of realism, we first calibrated the VIC model with the real observed streamflow from the USGS station to estimate these six parameters.

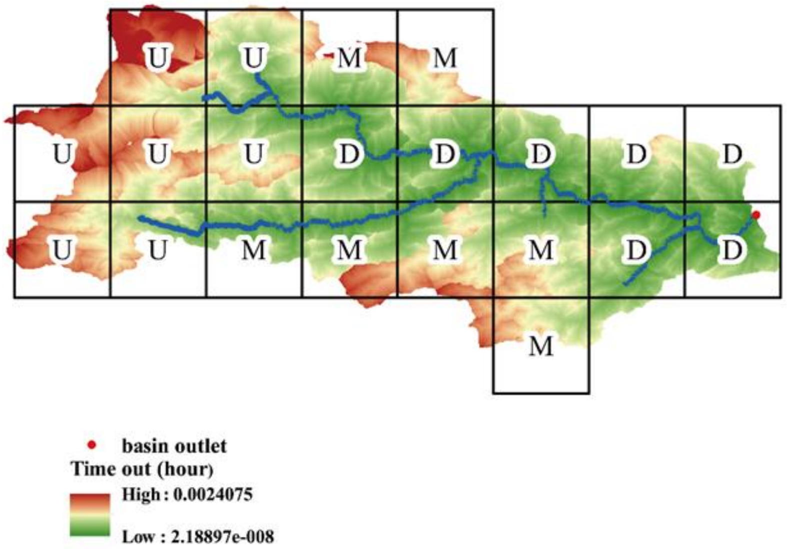


Figure 13 The French Creek Watershed Model

The map displays the water flow time to the outlet and the 21 VIC model cells with 1/32 degree resolution. The symbols in each cell represent a division of the watershed based on flow time. "D" denotes the downstream region; "M" denotes the midstream region; "U" denotes the upstream region.

The French Creek watershed has only two soil types according to the CONUS-SOIL dataset [68]: silt loam (92%) and loam (8%). To increase the dimensionality of the parameter space (given that DHSVM assigns parameters per soil type), and thus the complexity of the calibration problem, we randomly assigned each VIC model cell a unique soil texture, for a total of 21 textures. The textures were selected from the soil texture triangle defined by the U.S. Department of Agriculture (USDA). The white dots in Figure 14a show their locations in the triangle, which were selected for maximum heterogeneity. The corresponding soil parameter values for each VIC cell were identified based on the percentages of sand and clay. Regarding land cover, this watershed is dominated by deciduous forests (47%). We set the vegetation parameters to the default values based on the corresponding vegetation type. By using the six calibrated conceptual parameters with the synthetic soil texture, we obtained the synthetic "observed" streamflow from the VIC model.

Table 4 List of Conceptual Parameters in the VIC Modeling Engine

Parameter	Description	Typical range	Units	Calibrated value
b	Exponent of variable infiltration capacity curve	0-0.4	-	0.36
Ws	Fraction of maximum soil moisture content of the lowest layer where non-linear baseflow occurs	0-1.0	-	0.99
Dsmax	Maximum velocity of base flow	0-30.0	mm/d	1.63
Ds	Fraction of *Dsmax* where non-linear baseflow begins	0-1.0	-	0.88
d2	The depth of the second soil layer	0.1-2.0	m	0.17
d3	The depth of the third soil layer	0.1-2.0	m	0.11

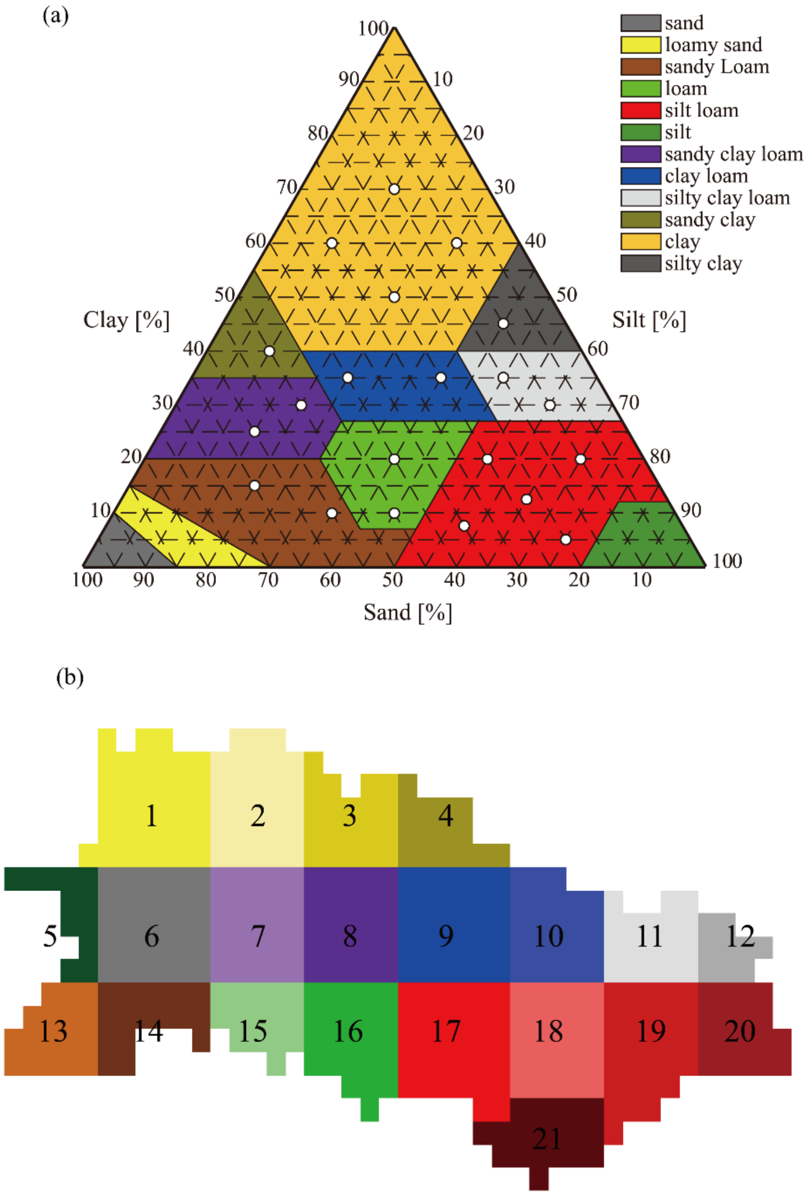


Figure 14 Synthetic Soil Type Assignments

(a) The soil texture triangle defined by the USDA. 21 soil textures (white dots) are selected to design the synthetic experiment. (b) Soil texture map of the watershed with resolution of the DHSVM model (500m). The ID numbers correspond to the 21 soil textures which are assigned to the 21 VIC model cells.

Afterwards, we set up the DHSVM model with the same soil texture and land cover type. The DHSVM model cells located within one VIC cell were assigned the corresponding soil texture of that VIC cell. Figure 14b shows the soil type distribution for the DHSVM model. The ID numbers correspond to the 21 soil textures which are assigned to the 21 VIC cells. We chose six sensitive parameters to be calibrated for each soil type in the DHSVM model. In addition to the four parameters that also exist in the VIC model, the other two parameters are the exponential decrease factor for the vertical hydraulic conductivity and the lateral saturated hydraulic conductivity.

As for vegetation parameters, we only calibrated some of the parameters of the dominating vegetation type (deciduous forest): the moisture threshold for the over-story and under-story, the aerodynamic attenuation, and the fractional coverage. Along with the general Manning's coefficient of the river channels, there is a total of 134 parameters which need to be calibrated in the DHSVM model. Among them, 87 parameters are also in the VIC model and thus have reference "true" values. Table 3 makes a detailed summary of the common soil parameters corresponding to the 21 soil textures. The soil parameter ranges of different soil texture types are provided by Meyer et al. (1997).

The rest of the parameters which need to be calibrated in the DHSVM model are shown in Table 6. Their corresponding ranges are based either on typical ranges applied in former studies or on their physical constraints. Given that the vertical saturated hydraulic conductivity is the least relevant parameter, as determined through the sensitivity analysis (shown in Table 6), and its parameter ranges span several orders of magnitude, we used a base-10 logarithmic scale for its range during the calibration process.

Table 5 List of Soil Parameters for Each Soil Texture in the DHSVM Model

Soil texture ID	Soil type[a]	Field capacity		Porosity		Wilting point		Vertical saturated hydraulic conductivity (m/s)	
		Range	"True" value	Range	"True" value	Range	"True" value	Range	"True" value
1	c	0.15-0.6	0.54	0.1-0.65	0.62	0.1-0.55	0.41	1E-8-1E-5	7.62E-7
2	c	0.15-0.6	0.45	0.1-0.65	0.56	0.1-0.55	0.34	1E-8-1E-5	4.14E-7
3	c	0.15-0.6	0.50	0.1-0.65	0.60	0.1-0.55	0.36	1E-8-1E-5	6.67E-7
4	c	0.15-0.6	0.42	0.1-0.65	0.54	0.1-0.55	0.28	1E-8-1E-5	4.80E-7
5	sc	0.15-0.5	0.32	0.25-0.55	0.50	0.12-0.35	0.22	1E-8-1E-5	4.14E-7
6	sic	0.15-0.55	0.42	0.15-0.6	0.56	0.1-0.4	0.26	1E-8-1E-5	7.70E-7
7	scl	0.1-0.45	0.25	0.2-0.6	0.47	0.07-0.2	0.15	1E-8-1E-5	1.17E-6
8	scl	0.1-0.45	0.28	0.2-0.6	0.48	0.07-0.2	0.17	1E-8-1E-5	8.11E-7
9	cl	0.1-0.6	0.32	0.13-0.65	0.50	0.07-0.35	0.20	1E-8-1E-5	6.67E-7
10	cl	0.1-0.6	0.35	0.13-0.65	0.52	0.07-0.35	0.19	1E-8-1E-5	8.69E-7
11	sicl	0.13-0.55	0.36	0.2-0.65	0.53	0.09-0.4	0.19	1E-7-1E-4	1.07E-6
12	sicl	0.13-0.55	0.34	0.2-0.65	0.53	0.09-0.4	0.17	1E-7-1E-4	1.64E-6
13	sl	0.05-0.3	0.21	0.13-0.6	0.44	0.02-0.15	0.11	1E-7-1E-4	3.69E-6
14	sl	0.05-0.3	0.22	0.13-0.6	0.41	0.02-0.15	0.09	1E-7-1E-4	7.33E-6
15	l	0.07-0.45	0.26	0.12-0.7	0.47	0.04-0.2	0.13	1E-7-1E-4	2.42E-6
16	l	0.07-0.45	0.24	0.12-0.7	0.43	0.04-0.2	0.09	1E-7-1E-4	7.42E-6
17	sil	0.05-0.5	0.31	0.2-0.7	0.49	0.03-0.3	0.12	1E-7-1E-3	3.69E-6
18	sil	0.05-0.5	0.28	0.2-0.7	0.45	0.03-0.3	0.10	1E-7-1E-3	6.50E-6
19	sil	0.05-0.5	0.29	0.2-0.7	0.48	0.03-0.3	0.12	1E-7-1E-3	2.92E-6
20	sil	0.05-0.5	0.26	0.2-0.7	0.41	0.03-0.3	0.09	1E-7-1E-3	9.89E-6
21	sil	0.05-0.5	0.30	0.2-0.7	0.38	0.03-0.3	0.09	1E-7-1E-3	1.16E-5

[a] The symbols of soil type in this table are acronyms of soil texture classifications defined by the USDA. The classifications appeared in the table include clay (c), sandy clay (sc), silty clay (sic), sandy clay loam (scl), clay loam (cl), silty clay loam (sicl), sandy loam (sl), loam (l), silt loam (sil).

Table 6 List of Additional DHSVM Parameters Which Need to be Calibrated

Parameter	Units	Range	Referenced "true" value from VIC
Soil parameters			
Lateral saturated hydraulic conductivity	m/s	1.0E-5-0.1	-
Exponent for change of lateral conductivity with depth (Exponential decrease)	-	0-10	-
Vegetation parameters			
Minimum stomatal resistance for the overstory	s/m	0-200	80
Minimum stomatal resistance for the understory	s/m	0-200	80
Radiation attenuation	-	0.1-0.8	0.5
Fractional coverage of overstory	-	0-1	-
Aerodynamic attenuation	-	0-3	-
Soil moisture threshold to restrict transpiration for the overstory	-	0-1	-
Soil moisture threshold to restrict transpiration for the understory	-	0-1	-
Routing parameter			
Manning's coefficient of the river channels	-	1.0E-6-0.1	-

4.3.2 Calibration Schemes

Following the steps for HIP-POP's strategy phase, we first conducted a sensitivity analysis using 2-level fractional factorial experiments [97] to determine the relative sensitivity rankings of the parameters which need to be calibrated. The results (not shown) were consistent with the conclusions of [164], which conducted a thorough sensitivity test of the DHSVM model. We then decided to divide the 134 parameters of the DHSVM model into $g = 6$ groups, with each group containing roughly equal number of parameters. Based on our hypothesis, the most sensitive parameters are to be assigned to the first groups so that they get more chances to be properly adjusted. Table 7 lists the grouping of parameters in our study. As can be seen, each of the top three groups contains field capacity, lateral saturated hydraulic conductivity and exponential decrease factor of seven types of soil texture, which correspond to seven VIC model cells/pixels. Because those three soil parameters are all very sensitive, and we were reluctant to rank them further in order of priority, we decided to put them into the top three groups and divide them by their corresponding cell in the VIC model. The problem of determining which of these cells to give priority to thus ensued. We therefore investigated three calibration configurations corresponding to three ways to divide these cells. The three configurations were compared with the traditional calibration scheme, in which all the parameters are calibrated simultaneously.

Table 7 List of Groups of the 134 DHSVM Model Parameters

Group	Parameters	Parameter count
1	field capacity (×7); lateral saturated hydraulic conductivity (×7); exponential decrease (×7); Manning's coefficient	22
2	field capacity (×7); lateral saturated hydraulic conductivity (×7); exponential decrease (×7)	21
3	field capacity (×7); lateral saturated hydraulic conductivity (×7); exponential decrease (×7)	21
4	porosity (×10); wilting point (×10); minimum stomatal resistance for overstory; minimum stomatal resistance for understory; fractional coverage of overstory	23
5	porosity (×11); wilting point (×11)	22
6	vertical saturated hydraulic conductivity (×21); radiation attenuation; aerodynamic attenuation; moisture threshold for overstory; moisture threshold for understory	25

We divided the watershed into three regions: downstream, midstream, and upstream, based on the flow time to the outlet (Figure 13). The three HIP-POP configurations and the traditional calibration scheme, which is to be used as a control, are described as follows:

1. *HIP-POP$_{D-U}$*. The soil parameters corresponding to the 7 VIC cells of the downstream region are in the first group, followed by midstream cells and upstream cells in the second and third group, respectively. For the fourth group, porosity and wilting point for 10 types of soil texture correspond to the 10 cells with top 10 shortest flow time, including 7 cells of the downstream region and 3 cells of the midstream region. The soil parameters corresponding to the remaining 11 cells are in the fifth group.

2. *HIP-POP$_{U-D}$*. Instead of calibrating parameters from downstream to upstream, *HIP-POP$_{U-D}$* reverses the calibration order of *HIP-POP$_{D-U}$*. That is, the soil parameters corresponding to the 7 VIC cells of the upstream region are in the first group, and the soil parameters for the downstream cells are in the third group. Similarly, the soil parameters for the 10 cells with top 10 longest flow times are in the fourth group.

3. *HIP-POP$_{Rand}$*. In this calibration scheme, the VIC cells are divided based on the ID numbers of soil texture in the DHSVM model (Figure 14b). Because each soil texture corresponds to only one VIC cell, the ID numbers can also represent VIC cells. Parameters for cells with ID 1-7 are in the first group, and so on. For porosity and wilting point, cells with ID 1-10 are in the fourth group.

4. *Trad.* All 134 parameters are calibrated together. This scheme represents traditional automatic calibration approaches and is therefore used as a control.

The setup for the three HIP-POP configurations can be seen in Table 3. The number of candidate solutions (or parameter combinations) to evaluate in each run n was set to increase with each one using the reasoning introduced in the previous section. The total number of allowed model evaluations was set to 4000 for all the schemes. The selected evolution of parameter ranges and number of discrete values for each group is also shown in Table 3.

4.3.3 Evaluation Metrics

We selected a set of metrics to evaluate the performance of the new calibration schemes based on the 20 best solutions/parameter combinations produced by each. Additionally, the Tukey boxplot [97] of each parameter was obtained from the 20 samples. These boxplots were used in the selection of the performance metrics. The bottom and top of the box correspond to the first (Q1) and third (Q3) quartiles, which are the 25th and 75th percentiles of the sample data, respectively. The interquartile range (IQR), which is a measure of statistical dispersion, is equal to the difference between the third and first quartiles (IQR = Q3 - Q1). For the Tukey boxplots, the lower whisker represents the lowest solution still within 1.5 IQR of the lower quartile (Q1 - 1.5×IQR), and the upper whisker represents the highest solution still within 1.5 IQR of the upper quartile (Q3 + 1.5×IQR). The statistic metrics are defined as follows:

1. *No convergence (NC)*. If the width between the two whiskers is greater than 10% of the original parameter range, it is considered that the parameter reached no convergence. *NC* is the total number of such parameters for a solution set. The non-convergence of parameters stems from equifinality and can mean that the algorithm was unable to find satisfactory assignments that perform consistently well. In this sense, a large NC number can be seen as a symptom of poor performance.

2. *Hitting the boundary (HB).* If the width between the median and the boundary of the parameter range is less than 1% of the original range, or the whiskers of the box reach the boundary, this parameter is considered to have hit the boundary. Realistic values of parameters are rarely located at the extremes of the variation intervals and, more often, convergence to the boundary usually means that the algorithm was seeking unrealistic values to compensate for failures in the assignments to other parameters. Therefore, a large HB number indicates poor performance.

3. Ratio of smaller *Absolute relative error (RosARE).* This metric is a relative score between the results of two different calibration schemes, specifically between a configuration of HIP-POP and the traditional method. Among the 87 parameters with referenced true values, the parameters that have converged in both the HIP-POP-based and the traditional schemes are selected. The absolute relative error (ARE) of each of these parameters is calculated using the reference true values. The RosARE is then computed as the number of parameters with smaller ARE in the HIP-POP-based scheme divided by the total number of parameters that converged. If the RosARE is higher than 50%, the HIP-POP configuration yielded more calibrated parameters that approach their “true” value than the traditional calibration scheme.

4.4 Results and Discussion

We first calibrated the VIC model with the observed streamflow from the USGS station at the basin outlet to an NSE of 0.6. The calibrated values of the six conceptual parameters are shown in Table 4. Figure 15 shows the simulated hydrograph using the optimized parameters and the observed hydrograph along with the areal precipitation time series. Although the NSE is not very high and the time series presents scattered underestimations of peak flows, the simulated hydrograph has the same trend as the observed hydrograph and the areal precipitation time series, with a good depiction of the baseflow. Therefore, the VIC model can be viewed as a reasonably-reliable tool to generate the synthetic "observed" streamflow in this watershed.

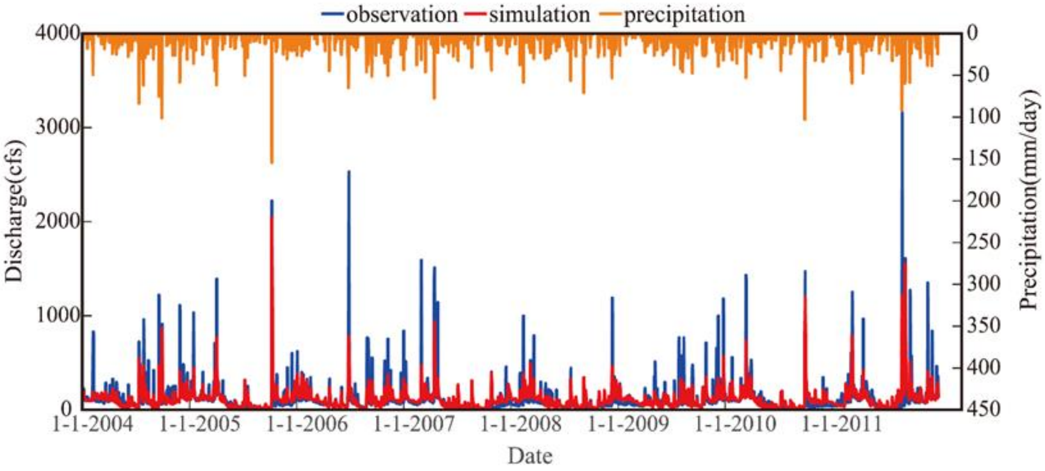


Figure 15 Hydrograph Comparison for the French Creek

The comparison is between the daily streamflow time series simulated by the VIC model and the observation. The calibration period is from 2004 to 2011, with 2003 used for spin-up.

Because of the stochastic nature of these calibration schemes, the relative performance of different calibration schemes must be assessed over multiple optimization trials. After each trial, we performed a one-way Analysis of Variance (ANOVA) for the NC and HB scores, and a 1-Sample t-test for the RosARE to determine the statistical significance of the difference between the HIP-POP-based calibration schemes and the traditional one. After only 3 trials we found that all the three configurations of HIP-POP produced statistically significantly fewer numbers of NC and HB than the traditional calibration scheme. It can be seen in Figure 16 that the Tukey's 95%

confidence intervals of differences of means for NC and HB between the HIP-POP configurations and the traditional one do not contain zero, indicating that the corresponding means are significantly different. The p-values of the two hypothesis tests for NC and HB are 0.004 and 0.006, which are smaller than the defined significance level of 0.05. In the 1-Sample t-test for RosARE, we assumed the hypothesized mean was 50% and set the significance level to 0.05. After 3 trials, only HIP-POP$_{D-U}$ produced a mean RosARE that was statistically significantly higher than 50% (p-value = 0.047).

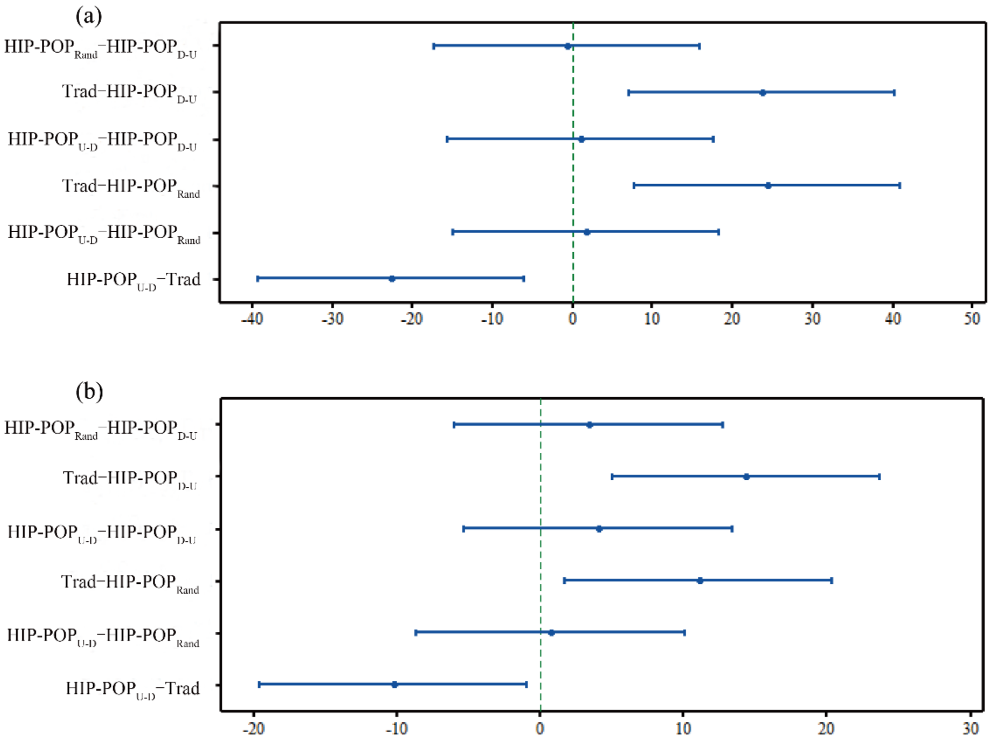


Figure 16 95% Confidence Intervals for a) NC and b) HB

The Tukey simultaneous 95% confidence intervals are of the differences of means for a) NC and b) HB. If one interval does not contain zero, the corresponding means are significantly different.

Table 8 Comparison of the Different Calibration Schemes

	HIP-POP$_{D-U}$	HIP-POP$_{U-D}$	HIP-POP$_{Rand}$	Trad
NC	7	8.3	6.7	31
HB	8.3	12.3	11.7	22.7
RosARE	58.8%	49.6%	48.7%	-

The means of the three evaluation metrics of the three trials are listed in Table 8. The three HIP-POP configurations have significantly better performance than Trad in terms of NC and HB. The number of not converged parameters for Trad is nearly three times those of the HIP-POP configurations. It can be seen in Table 9 that most of the parameters that did not converge with HIP-POP are in the last group, which had the lowest chances to be adjusted. However, the parameters that did not converge for Trad are distributed throughout all groups. Among the three HIP-POP configurations, HIP-POP $_{D-U}$ has the least number of parameters hitting the boundary and produces the largest number of parameters that are close to their "true" values. The reason why the HIP-POP$_{D-U}$ configuration produces the best result is probably related to the fact that the surface runoff from the downstream region arrives at the basin outlet the earliest and, therefore, the soil parameters for this region are likely to have the most impact on the streamflow at the outlet. For the upstream region, the surface runoff from these cells would spend a longer period of time traveling to the outlet. A number of factors play a role in affecting the streamflow at the outlet through the routing process. Hence, the soil parameters for this region appear to have a less direct impact on the streamflow at the outlet.

Table 9 Number of Not Converged Parameters Within Each Parameter Group

	Group1	Group2	Group3	Group4	Group5	Group6
Grouping based on HIP-POP$_{D\text{-}U}$						
HIP-POP$_{\text{D-U}}$				0.7		6.3
Trad	3.7	3	2.7	8.3	5.3	8
Grouping based on HIP-POP$_{U\text{-}D}$						
HIP-POP$_{\text{U-D}}$			0.3		2.3	5.7
Trad	2	3	4.3	5.7	8	8
Grouping based on HIP-POP$_{Rand}$						
HIP-POP$_{\text{Rand}}$					1	5.7
Trad	1.7	4.7	3	6.7	7	8

When parameters hit the upper or lower boundary of the predefined range, this could be an indication that the parameter values might need to be re-calibrated, and a widening of the parameter ranges could be in order. However, some of the ranges cannot be further widened without violating physical constraints. The interactions between different hydrological processes may lead to poor parameter identifiability because some parameters compensate for others to make the simulated hydrograph imitate the observed one. When all the parameters are calibrated together, there are more chances that some parameters are affected by others in the high-dimensional parameter space. In addition, the presence of different sources of uncertainty, e.g., from the model structure, may also lead to unrealistic parameter estimations. Therefore, the parameters calibrated with Trad cannot represent the real behavior of the basin. In contrast, calibration schemes based on HIP-POP, especially HIP-POP$_{\text{D-U}}$, can partially overcome the effect of model structure error and mitigate the severe equifinality, thus providing much more reliable parameter estimations. This result stands even in cases where the hydrograph fit, as measured with the NSE, would be higher for Trad; as these mechanisms in HIP-POP mitigate the problem of being “right for the wrong reasons.”

Finally, Figure 17 shows the evolution of the best NSE obtained for the three optimization trials. Overall, the calibration schemes based on HIP-POP are able to more rapidly and more consistently increase the NSE. The traditional approach displays a much slower convergence speed in two out of the three cases, and sudden jumps followed by stagnancy periods. We attribute this behavior to the larger solution space being navigated, where finding combinations that increase the NSE is harder. Progress in the HOP-POP-based schemes is easier as the size of the solutions space is much smaller during the entire calibration process and, thus, parameter assignments can be explored more individually and taking into account those with larger impact first. This observed behavior therefore make HIP-POP more suitable for parameter calibrations with high-dimensional parameter space that are associated with computationally intensive hydrologic models.

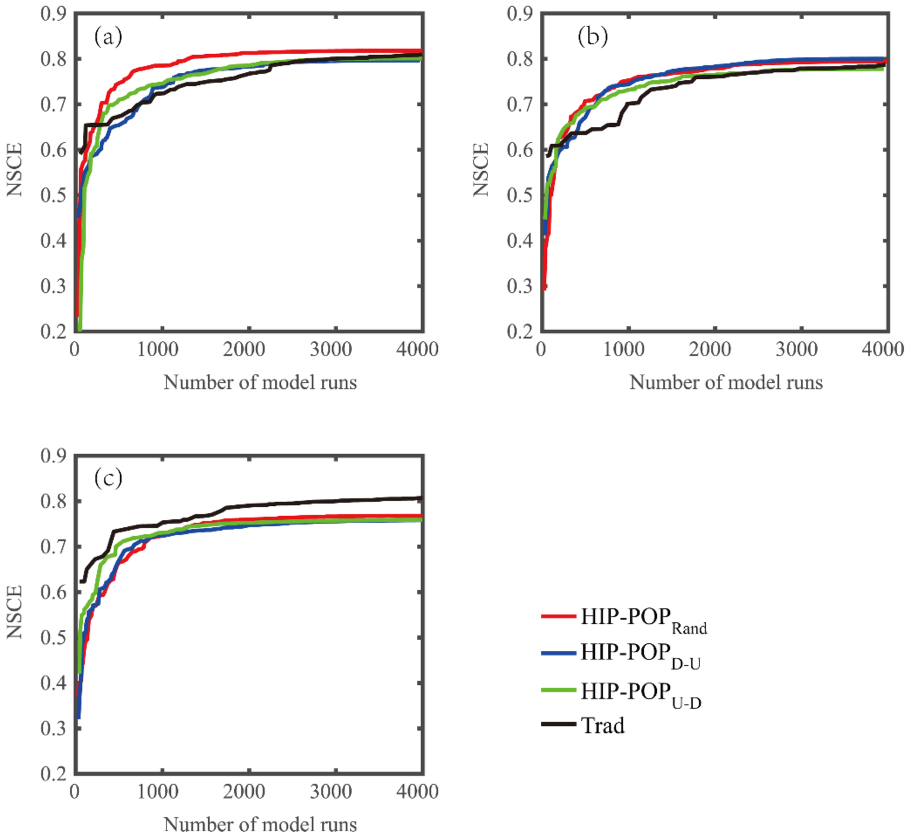


Figure 17 Evolution of the NSE for Three Optimization Trials.

4.5 Conclusions and Future Work

This study introduced HIP-POP, a hybrid calibration methodology for high-resolution hydrologic models that incorporates expert-knowledge principles into a global optimization framework. HIP-POP divides the calibration problem into several runs of the optimization algorithm and focuses on a different group of unknown parameters in each of them in a systematic way. This approach not only enables giving priority to the most sensitive parameters, but also allows for an iterative and gradual reduction of the uncertainty and for a reduction in the computational demands of the optimization algorithm by reducing the size of the search space in each run.

Different configurations of HIP-POP were tested in a quasi-synthetic experiment which involved two well-known modeling engines: VIC and DHSVM. A VIC model for the French Creek watershed in Pennsylvania was calibrated to match the observed streamflow at the outlet. Then a modified version of the VIC model was used to serve as the synthetic ground truth to calibrate another version of the model using the DHSVM. This double-model setup manages to provide much more control over the "ground truth" compared to using direct measurements and, at the same time, incorporates the structural uncertainties of models that are neglected in fully-synthetic experimental setups in previous studies in the literature.

The tests compared the configurations of HIP-POP with a traditional optimization-based algorithm based on several metrics: the final Nash-Sutcliffe coefficient, the rate of improvement, the ratio of parameters that converged, the ratio of parameters that hit the boundary of the allowed range, and the difference between the optimized values and the available "true" values. While the HIP-POP methodology did not always produce better NSE values, it proved to have an accelerated improvement rate, higher ratios of converged and non-boundary parameter values, and smaller differences with the synthetic target parameter values. These results demonstrate HIP-POP is advantageous in tackling calibration problems of high dimensionality both because of its smaller reliance on numerous model evaluations and because of its ability to mitigate the problems related to equifinality; that is, due to its increased capabilities in discerning more realistic solutions despite the availability of multiple high-performing parameter combinations.

Future work should explore multiple avenues of improvement for the HIP-POP framework. First, we would like to investigate the performance of the framework with different types of global optimization algorithms. Second, extending the calibration framework to optimize multiple objective simultaneously or using a formal Bayesian approach [160], [165] could further help in reducing the uncertainty in high-resolution models. Finally, some studies have pointed out that parameters associated with subsurface fluxes get poorly optimized when traditional model calibration approaches, using observed streamflow, were used [72], [166], [167]. We intend to test if calibration based on HIP-POP can further lead to better estimation of these subsurface fluxes.

5.0 Model Initialization through Hybridized Data Assimilation

This chapter moves from parameter uncertainty to initial state uncertainty, and introduces a hybrid data assimilation algorithm that allows for the probabilistic estimation of the initial conditions in watersheds. The chapter is based on the article “Hybridizing Bayesian and variational data assimilation for robust high-resolution hydrologic forecasting” by Felipe Hernández and Xu Liang, published in the Hydrology and Earth System Sciences journal of the European Geophysical Union in 2018 [168]. The article is distributed under the Creative Commons Attribution 4.0 License and, thus, can be reproduced here provided proper attribution.

5.1 Introduction

There exists a plethora of techniques to initialize the state variables of a model through the incorporation of available observations, and they possess overlapping features that make it difficult to develop clear-cut classifications. However, two main “schools” can be fairly identified: Bayesian data assimilation and variational data assimilation. Bayesian data assimilation creates probabilistic estimates of the state variables in an attempt to also capture their uncertainty. These state probability distributions are adjusted sequentially to better match the observations using Bayes’ theorem. While the Kalman filter (KF) is constrained to linear dynamics and Gaussian distributions, ensemble Kalman filters (EnKF) can support non-linear models [169], and particle filters (PF) can also manage non-Gaussian estimates for added accuracy [170]. The stochastic nature of these Bayesian filters is highly valuable because equifinality can rarely be avoided and because of the benefits of quantifying uncertainty in forecasting applications [171], [172]. While superior in accuracy, PFs are usually regarded as impractical for high-dimensional applications [173], and thus recent research has focused on improving their efficiency [174].

On the other hand, variational data assimilation is more akin to traditional calibration approaches because of its use of optimization methods. It seeks to find a single/deterministic initial state variable combination that minimizes the departures (or "variations") of the modelled values from the observations [175] and, commonly, from their history. One- to three- dimensional variants are also employed sequentially, but the paradigm lends itself easily to evaluating the performance of candidate solutions throughout an extended time window in four-dimensional versions (4D-Var). If the model's dynamics are linearized, the optimum can be very efficiently found in the resulting convex search space through the use of gradient methods. While this feature has made 4D-Var very popular in meteorology and oceanography [176], its application in hydrology has been less widespread because of the difficulty of linearizing land-surface physics [177]. Moreover, variational data assimilation requires the inclusion of computationally-expensive adjoint models if one wishes to account for the uncertainty of the state estimates [178].

Traditional implementations from both schools have interesting characteristics and thus the development of hybrid methods has received considerable attention [179]. For example, Bayesian filters have been used as adjoints in 4D-Var to enable probabilistic estimates [180]. Moreover, some Bayesian approaches have been coupled with optimization techniques to select ensemble members [181], [182]. 4DEnVar [183], a fully-hybridized algorithm, is gaining increasing attention for weather prediction [184], [185]. It is especially interesting that some algorithms have defied the traditional choice between sequential and "extended-time" evaluations. Weak-constrained 4D-Var allows state estimates to be determined at several time steps within the assimilation time window and not only at the beginning [186], [187]. Conversely, modifications to EnKFs and PFs have been proposed to extend the analysis of candidate members/particles to span multiple time steps [188], [189]. The success of these hybrids demonstrates that there is a balance to be sought between the allowed number of degrees of freedom and the amount of information to be assimilated at once.

Following these promising paths, in this chapter we introduce OPTIMISTS (Optimized PareTo Inverse Modeling through Integrated STochastic Search), a hybrid data assimilation algorithm whose design was guided by two goals: to allow for practical scalability to high-dimensional models, and to enable balancing the imperfect observations and the imperfect model estimates to minimize overfitting.

Table 10 summarizes the main characteristics of typical Bayesian and variational approaches, and their contrasts with those of OPTIMISTS. Our algorithm incorporates the features that the literature has found to be the most valuable from both Bayesian and variational methods while mitigating the deficiencies or disadvantages associated with these original approaches (e.g., the linearity and determinism of 4D-Var and the limited scalability of PFs): Non-Gaussian probabilistic estimation and support for non-linear model dynamics have been long held as advantageous over their alternatives [190], [191] and, similarly, meteorologists favour extended-period evaluations over sequential ones [192]–[194]. As shown in the table, OPTIMISTS can readily adopt these proven strategies.

Table 10 Comparison Between OPTIMISTS and Standard Data Assimilation Algorithms

Included algorithms: Bayesian (KF: Kalman Filter, EnKF: Ensemble KF, PF: Particle Filter), variational (one- to four-dimensional), and OPTIMISTS.

	Bayesian	Variational	OPTIMISTS
Resulting state-variable estimate	Probabilistic: Gaussian (KF, EnKF), Non-Gaussian (PF)	Deterministic (unless adjoint model is used)	Probabilistic (using kernel density estimation)
Solution quality criteria	High likelihood given observations	Minimum cost value (error, departure from history)	Flexible: e.g., min. error, max. consistency with history
Analysis time step	Sequential	Sequential (1D-3D) or entire assimilation window (4D)	Flexible
Search method	Iterative Bayesian belief propagation	Convex optimization	Coupled belief propagation/multi-objective optimization
Model dynamics	Linear (KF), non-linear (EnKF, PF)	Linearized to obtain convex solution space	Non-linear (non-convex solution space)

However, there are other aspects of the assimilation problem for which no single combination of features has demonstrated its superiority. For example, is the consistency with previous states better achieved through the minimization of a cost function that includes a background error term [195], as in variational methods, or through limiting the exploration to samples drawn from that background state distribution, as in Bayesian methods?

Table 10 shows that in these cases OPTIMISTS allows for flexible configurations, and it is an additional objective of this study to test which set of feature interactions allows for more accurate forecasts when using highly-distributed models. While many of the concepts utilized within the algorithm have been proposed in the literature before, their combination and broad range of available configurations are unlike those of other methods, including existing hybrids which have mostly been developed around ensemble Kalman filters and convex optimization techniques [179]—and therefore limited to Gaussian distributions and linear dynamics.

5.2 Data Assimilation Algorithm

In this section we describe OPTIMISTS, our proposed data assimilation algorithm which combines advantageous features from several Bayesian and variational methods. As will be explained in detail for each of the steps of the algorithm, these features were selected with the intent of mitigating the limitations of existing methods.

The objective of data assimilation is to find an adequate estimate of the starting or initial state variables $\boldsymbol{s}^{t_i}$ for a forecasting simulation between t_i and t_f given one or more observations (of output or state variables) right until t_i. For a semi-distributed or fully distributed hydrologic model, state variable vector $\boldsymbol{s}$ is the concatenation of the state variables in each of the model's elements l (e.g., sub-watersheds or pixels/cells, channel reaches, etc.):

$$\boldsymbol{s} = \{\boldsymbol{s}^l \ \forall \ l \in \boldsymbol{\mathcal{M}}\} \tag{5}$$

OPTIMISTS allows selecting a flexible data assimilation time step Δt—i.e., the time window in which candidate state configurations are compared to observations. It can be as short as the model time step, or as long as the entire assimilation window. For each assimilation time step at time t a new state probability distribution $\boldsymbol{S}^{t+\Delta t}$ is estimated from the current distribution $\boldsymbol{S}^t$, the model, and one or more observations $\boldsymbol{o}_{\text{obs}}^{t:t+\Delta t}$. For hydrologic applications, as those explored in this article, these states $\boldsymbol{S}$ include land surface variables within the modelled watershed such as soil moisture, snow cover/water equivalent, and stream water volume; and observations $\boldsymbol{o}$ are typically of streamflow at the outlet [196], soil moisture [197], [198], and/or snow cover [199]. However, the description of the algorithm will use field-agnostic terminology not to discourage its application in other disciplines.

State probability distributions $\boldsymbol{S}$ in OPTIMISTS are determined from a set of weighted "root" or "base" sample states $\boldsymbol{s}_i$ using multivariate weighted kernel density estimation [200]. This form of non-parametric distributions stands in stark contrast with those from KFs and EnKFs in their ability to model non-Gaussian behaviour—an established advantage of PFs. Each of these samples or ensemble members $\boldsymbol{s}_i$ is comprised of a value vector for the state variables. The objective of the algorithm is then to produce a set of n samples $\boldsymbol{s}_i^{t+\Delta t}$ with corresponding weights w_i for the next assimilation time step to determine the target distribution $\boldsymbol{S}^{t+\Delta t}$.

This process is repeated iteratively each assimilation time step Δt until the entire assimilation time frame is covered, at which point the resulting distribution can be used to perform the forecast simulations. In Subsection 4.2.1 we describe the main ideas and steps involved in the OPTIMISTS data assimilation algorithm; details regarding the state probability distributions, mainly on how to generate random samples and evaluate the likelihood of particles, are explained in Subsection 4.2.2; and modifications required for high-dimensional problems are described in Subsection 4.2.3.

5.2.1 Description of the OPTIMISTS Data Assimilation Algorithm

Let a "particle" $\boldsymbol{P}_i$ be defined by a "source" (or initial) vector of state variables $\boldsymbol{s}_i^t$ (which is a sample of distribution $\boldsymbol{S}^t$), a corresponding "target" (or final) state vector $\boldsymbol{s}_i^{t+\Delta t}$ (a sample of distribution $\boldsymbol{S}^{t+\Delta t}$), a set of output values $\boldsymbol{o}_i^{t:t+\Delta t}$ (those that have corresponding observations $\boldsymbol{o}_{\text{obs}}^{t:t+\Delta t}$), a set of fitness metrics $\boldsymbol{f}_i$, a rank r_i, and a weight w_i. Note that the denomination "particle" stems from the PF literature and is analogous to the "member" term in EnKFs. The fitness metrics $\boldsymbol{f}_i$ are used to compare particles with each other in the light of one or more optimization objectives.

The algorithm consists of the following steps, whose motivation and details are included in the subsubsections below and their interactions illustrated in Figure 18. Table 11 lists the meaning of each of the seven global parameters (Δt, n, w_{root}, p_{samp}, $k_{\text{F-class}}$, n_{evo}, and g).

1. Drawing: draw root samples $\boldsymbol{s}_i^t$ from $\boldsymbol{S}^t$ in descending weight order until $\sum w_i \geq w_{\text{root}}$
2. Sampling: randomly sample $\boldsymbol{S}^t$ until the total number of samples in the ensemble is $p_{\text{samp}} \times n$
3. Simulation: compute $\boldsymbol{s}_i^{t+\Delta t}$ and $\boldsymbol{o}_i^{t:t+\Delta t}$ from each non-evaluated sample $\boldsymbol{s}_i^t$ using the model
4. Evaluation: compute the fitness values $\boldsymbol{f}_i$ for each particle $\boldsymbol{P}_i$
5. Optimization: create additional samples using evolutionary algorithms and return to 3 (if number of samples is below n)
6. Ranking: assign ranks r_i to all particles $\boldsymbol{P}_i$ using non-dominated sorting
7. Weighting: compute the weight w_i for each particle $\boldsymbol{P}_i$ based on its rank r_i

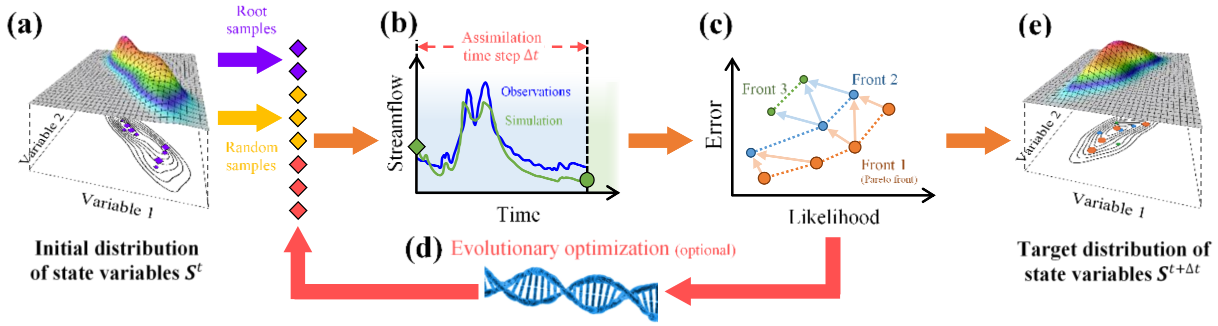

(a)
Variable 2
Variable 1
Initial distribution of state variables S^t
Root samples
Random samples
(b)
Assimilation time step Δt
Streamflow
Observations
Simulation
Time
(c)
Error
Front 3
Front 2
Front 1
(Pareto front)
Likelihood
(e)
Variable 2
Variable 1
Target distribution of state variables $S^{t+\Delta t}$
(d) Evolutionary optimization (optional)

Figure 18 Steps in OPTIMISTS, to be Repeated for Each Assimilation Time Step Δt

In this example state vectors have two variables, observations are of streamflow, and particles are judged using two user-selected objectives: the likelihood given $\boldsymbol{S}^t$ to be maximized and the error given the observations to be minimized. (a) Initial state kernel density distribution $\boldsymbol{S}^t$ from which root samples (purple rhombi) are taken during the drawing step and random samples (yellow rhombi) are taken during the sampling step. (b) Execution of the model (simulation step) for each source sample for a time equal to Δt to compute output variables (for comparison with observations) and target samples (circles). (c) Evaluation of each particle (evaluation step) based on the objectives and organization into non-domination fronts (ranking step). The dashed lines represent the fronts while the arrows denote domination relationships between

particles in adjacent fronts. (d) Optional optimization step which can be executed several times and that uses a population-based evolutionary optimization algorithm to generate additional samples (red rhombi). (e) Target state kernel density distribution $\boldsymbol{S}^{t+\Delta t}$ constructed from the particles' final samples (circles) after being weighted according to the rank of their front (weighting step): kernels centred on samples with higher weight (shown larger) have a higher probability density contribution.

Table 11 List of Global Parameters in OPTIMISTS

Symbol	Description	Range
Δt	Assimilation time step (particle evaluation time frame)	$\mathbb{R}^+$
n	Total number of root states $\boldsymbol{s}_i$ in the probability distributions	$\mathbb{N} \geq 2$
w_{root}	Total weight of root samples drawn from $\boldsymbol{S}^t$	$\mathbb{R} \in [0, 1]$
p_{samp}	Percentage of n corresponding to drawn and random samples	$\mathbb{R} \in [0, 1]$
$k_{\text{F}-\text{class}}$	Whether or not to use F-class kernels. If not: D-class kernels.	true or false
n_{evo}	Samples to be generated by the optimizers per iteration	$\mathbb{N} \geq 2$
g	Level of greed for the assignment of particle weights w_i	$\mathbb{R} \in [-1, 1]$

5.2.1.1 Drawing Step

While traditional PFs draw all the root (or base) samples from $\boldsymbol{S}^t$ [190], OPTIMISTS can limit this selection to a subset of them. The root samples with the highest weight—those that are the "best performers"—are drawn first, then the next ones in descending weight order, until the total weight of the drawn samples $\sum w_i$ reaches w_{root}. w_{root} thus controls what percentage of the root samples to draw, and, if set to one, all of them are selected.

5.2.1.2 Sampling Step

In this step the set of root samples drawn is complemented with random samples. The distinction between root samples and random samples is that the former are those that define the probability distribution $\boldsymbol{S}^t$ (that serve as centroids for the kernels), while the latter are generated stochastically from the kernels. Random samples are generated until the size of the combined set reaches $p_{\text{samp}} \times n$ by following the equations introduced in Subsection 4.2.2. This second step contributes to the diversity of the ensemble in order to avoid sample impoverishment as seen on PFs [201], and serves as a replacement for traditional resampling strategies [202]. The parameter w_{root} therefore controls the intensity with which this feature is applied to offer users some level of flexibility. Generating random samples at the beginning, instead of resampling those that have been already evaluated, could lead to discarding degenerate particles (those with high errors) early on and contribute to improved efficiency, given that the ones discarded are mainly those with the lowest weight as determined in the previous assimilation time step.

5.2.1.3 Simulation Step

In this step, the algorithm uses the model to compute the resulting state vector $\boldsymbol{s}_i^{t+\Delta t}$ and an additional set of output variables $\boldsymbol{o}_i^{t:t+\Delta t}$ for each of the samples (it is possible that state variables double as output variables). The simulation is executed starting at time t for the duration of the assimilation time step Δt (not to be confused with the model time step which is usually shorter). Depending on the complexity of the model, the simulation step can be the one with the highest computational requirements. In those cases, parallelization of the simulations would greatly help in reducing the total footprint of the assimilation process. The construction of each particle $\boldsymbol{P}_i$ is started by assembling the corresponding values computed so far: $\boldsymbol{s}_i^t$ (drawing, sampling, and optimization steps), and $\boldsymbol{s}_i^{t+\Delta t}$ and $\boldsymbol{o}_i^{t:t+\Delta t}$ (simulation step).

5.2.1.4 Evaluation Step

In order to determine which initial state $\boldsymbol{s}_i^t$ is the most desirable, a two-term cost function J is typically used in variational methods that simultaneously measures the resulting deviations of modelled values $\boldsymbol{o}_i^{t:t+\Delta t}$ from observed values $\boldsymbol{o}_{\text{obs}}^{t:t+\Delta t}$ and the departures from the background state distribution $\boldsymbol{S}^t$ [195]. The function usually has the form shown in Eq. (6):

$$J_i = c_1 \cdot J_{\text{background}}(\boldsymbol{s}_i^t, \boldsymbol{S}^t\ ) + c_2 \cdot J_{\text{observations}}\left(\boldsymbol{o}_i^{t:t+\Delta t}, \boldsymbol{o}_{\text{obs}}^{t:t+\Delta t}\right), \tag{6}$$

where c_1 and c_2 are balancing constants usually set so that $c_1 = c_2$. Such a multi-criteria evaluation is crucial both to guarantee a good level of fit with the observations (second term) and to avoid the optimization algorithm to produce an initial state that is inconsistent with previous states (first term)—which could potentially result in overfitting problems rooted in disproportionate violations of mass and energy conservation laws (e.g., in hydrologic applications a sharp, unrealistic rise in the initial soil moisture could reduce $J_{\text{observations}}$ but would increase $J_{\text{background}}$). In Bayesian methods, since the consistency with the history is maintained by sampling only from the prior/background distribution $\boldsymbol{S}^t$, single term functions are used instead—which typically measure the probability density or likelihood of the modelled values given a distribution of the observations.

In OPTIMISTS any such fitness metric could be used and, most importantly, the algorithm allows defining several of them. Moreover, users can determine whether if each function is to be minimized (e.g., costs or errors) or maximized (e.g., likelihoods). We expect these features to be helpful if one wishes to separate errors when multiple types of observations are available [203] and as a more natural way to consider different fitness criteria (lumping them together in a single function as in Eq. (6) can lead to balancing and “apples and oranges” complications). Moreover, it might prove beneficial to take into account the consistency with the state history both by explicitly defining such an objective here and by allowing states to be sampled from the previous distribution (and thus compounding the individual mechanisms of Bayesian and variational methods). Functions to measure this consistency are proposed in Subsection 4.2.2. With the set of objective functions defined by the user, the algorithm computes the vector of fitness metrics $\boldsymbol{f}_i$ for each particle during the evaluation step.

5.2.1.5 Optimization Step

The optimization step is optional and is used to generate additional particles by exploiting the knowledge encoded in the fitness values of the current particle ensemble. In a twist to the signature characteristic of variational data assimilation, OPTIMISTS incorporates evolutionary multi-objective optimization algorithms [156] instead of the established gradient-based, single-objective methods. Evolutionary optimizers compensate their slower convergence speed with the capability of efficiently navigating non-convex solution spaces (i.e., the models and the fitness functions do not need to be linear with respect to the observations and the states). This feature effectively opens the door for variational methods to be used in disciplines where the linearization of the driving dynamics is either impractical, inconvenient, or undesirable. Whereas any traditional multi-objective global optimization method would work, our implementation of OPTIMISTS features the adaptive ensemble algorithm introduced in Chapter 3.

During the optimization step, the group of optimizers is used to generate n_{evo} new sample states $\boldsymbol{s}_i^t$ based on those in the current ensemble. For example, the genetic algorithm selects pairs of base samples with high performance scores $\boldsymbol{f}_i$ and then proceeds to combine their individual values using standard crossover and mutation operators. The simulation and evaluation steps are repeated for these new samples, and then this iterative process is repeated until the particle ensemble has a size of n. Note that w_{root} and p_{samp} thus determine what percentage of the particles is generated in which way. For example, for relatively small values of w_{root} and a p_{samp} of 0.2, 80% of the particles will be generated by the optimization algorithms. In this way, OPTIMISTS offers its users the flexibility to behave anywhere in the range between "fully Bayesian" ($p_{\mathrm{samp}} = 1$) and "fully variational" ($p_{\mathrm{samp}} = 0$) in terms of particle generation. In the latter case, in which no root and random samples are available, the initial "population"/ensemble of states $\boldsymbol{s}_i^t$ is sampled uniformly from the viable range of each state variable.

5.2.1.6 Ranking Step

A fundamental aspect of OPTIMISTS is the way in which it provides a probabilistic interpretation to the results of the multi-objective evaluation, thus bridging the gap between Bayesian and variational assimilation. Such method has been used before [204] and is based on the employment of non-dominated sorting [156], another technique from the multi-objective optimization literature, which is used to balance the potential tensions between various objectives. This sorting approach is centered on the concept of "dominance," instead of organizing all particles from the "best" to the "worst." A particle dominates another if it outperforms it according to at least one of the criteria/objectives while simultaneously is not outperformed according to any of the others. Following this principle, in the ranking step particles are grouped in "fronts" comprised of members which are mutually non-dominated; that is, none of them is dominated by any of the rest. Particles in a front, therefore, represent the effective trade-offs between the competing criteria.

Figure 18.c illustrates the result of non-dominated sorting applied to nine particles being analyzed under two objectives: minimum deviation from observations and maximum likelihood given the background state distribution $\boldsymbol{S}^t$. Note that if a single objective function is used, the sorting method assigns ranks from best to worst according to that function, and two particles would only share ranks if their fitness values coincide. In our implementation we use the fast non-dominated sorting algorithm to define the fronts and assign the corresponding ranks r_i [83]. More efficient non-dominated sorting alternatives are available if performance becomes an issue [205].

5.2.1.7 Weighting Step

In this final step, OPTIMISTS assigns weights w_i to each particle according to its rank r_i as shown in Eqs. (7) and (8). This Gaussian weighting depends on the ensemble size n and the greed parameter g, and is similar to the one proposed by [87]. When g is equal to zero, particles in all fronts are weighted uniformly; when g is equal to one, only particles in the Pareto/first front are assigned non-zero weights. With this, the final estimated probability distribution of state variables for the next time step $\boldsymbol{S}^{t+\Delta t}$ can be established using multivariate weighted kernel density estimation (details in the next subsection), as demonstrated in Figure 18.e, by taking all target states $\boldsymbol{s}_i^{t+\Delta t}$ (circles) as the centroids of the kernels. The obtained distribution $\boldsymbol{S}^{t+\Delta t}$ can then be used as the initial distribution for a new assimilation time step or, if the end of the assimilation window has been reached, it can be used to perform (ensemble) forecast simulations.

$$w_i = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(r_i-1)^2}{2\sigma^2}} \tag{7}$$

$$\sigma = n \cdot [0.1 + 9.9 \cdot (1 - g)^5] \tag{8}$$

5.2.2 Model State Probability Distributions

As mentioned before, OPTIMISTS uses kernel density probability distributions [200] to model the stochastic estimates of the state variable vectors. The algorithm requires two computations related to the state-variable probability distribution $\boldsymbol{S}^t$: obtaining the probability density p or likelihood $\mathcal{L}$ of a sample and generating random samples. The first computation can be used in the evaluation step as an objective function to preserve the consistency of particles with the state history (e.g., to penalize aggressive departures from the prior conditions). It should be noted that several metrics that try to approximate this consistency exist, from very simple [204] to quite complex [187]. For example, it is common in variational data assimilation to utilize the background error term

$$J_{\text{background}} = (\boldsymbol{s} - \boldsymbol{s}_b)^{\mathrm{T}} \mathbf{C}^{-1} (\boldsymbol{s} - \boldsymbol{s}_b), \tag{9}$$

where $\boldsymbol{s}_b$ and $\mathbf{C}$ are the mean and the covariance of the "background" state distribution ($\boldsymbol{S}^t$ in our case) which is assumed to be Gaussian [195]. The term $J_{\text{background}}$ is plugged into the cost function shown in Eq. (6). For OPTIMISTS, we propose that the probability density of the weighted state kernel density distribution $\boldsymbol{S}^t$ at a given point (p) be used as a stand-alone objective. The density is given by Eq. (10) [206]. If Gaussian kernels are selected, the kernel function K, parameterized by the bandwidth matrix $\mathbf{B}$, is evaluated using Eq. (11).

$$p(\boldsymbol{s}|\boldsymbol{S}) = \frac{1}{\sum w_i} \sum_{i=1}^{n} [w_i \cdot K_{\mathbf{B}}(\boldsymbol{s} - \boldsymbol{s}_i)] \tag{10}$$

$$K_{\mathbf{B}}^{\text{Gauss}}(\boldsymbol{z}) = \frac{1}{\sqrt{(2\pi)^n \cdot |\mathbf{B}|}} \exp\left(-\frac{1}{2} \boldsymbol{z}^{\mathrm{T}} \mathbf{B}^{-1} \boldsymbol{z}\right) \tag{11}$$

Matrix $\mathbf{B}$ is the covariance matrix of the kernels, and thus determines their spread and orientation in the state space. $\mathbf{B}$ is of size $d \times d$, where d is the dimensionality of the state distribution (i.e., the number of variables), and can be thought of as a scaled-down version of the "background error covariance" matrix $\mathbf{C}$ from the variational literature. In this sense, matrix **B**, together with the spread of the ensemble of samples $\boldsymbol{s}_i$, effectively encode the uncertainty of the state variables. Several optimization-based methods exist to compute **B** by attempting to minimize the asymptotic mean integrated squared error (AMISE) [207], [208]. However, here we opt to use a simplified approach for the sake of computational efficiency: we determine **B** by scaling down the sample covariance matrix $\mathbf{C}$ using Silverman's rule of thumb, which takes into account the number of samples n and the dimensionality of the distribution d, as shown in Eq. (12) [88]. Figure 18 shows the density of two two-dimensional example distributions using this method (a and e). If computational constraints are not a concern, using AMISE-based methods or kernels with variable bandwidth [209], [210] could result in higher accuracy.

$$\mathbf{B}^{\text{Silverman}} = \left(\frac{4}{d+2}\right)^{\frac{2}{d+4}} \cdot n^{-\frac{2}{d+4}} \cdot \mathbf{C} \tag{12}$$

Secondly, OPTIMISTS' sampling step requires generating random samples from a multivariate weighted kernel density distribution. This is achieved by dividing the problem into two: we first select the root sample and then generate a random sample from the kernel associated with that base sample. The first step corresponds to randomly sampling a multinomial distribution with n categories and assigning the normalized weights of the particles as the probability of each category. Once a root sample $\boldsymbol{s}_{\text{root}}$ is selected, a random sample $\boldsymbol{s}_{\text{random}}$ can be generated from a vector $\boldsymbol{v}$ of independent standard normal random values of size d and a matrix **A** as shown in Eq. (13). **A** can be computed from a Cholesky decomposition [211] such that $\mathbf{A}\mathbf{A}^{\mathrm{T}} = \mathbf{B}$. Alternatively, an eigendecomposition can be used to obtain $\mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^{\mathrm{T}} = \mathbf{B}$ to then set $\mathbf{A} = \mathbf{Q}\boldsymbol{\Lambda}^{½}$.

$$\boldsymbol{s}_{\text{random}} = \boldsymbol{s}_{\text{root}} + \mathbf{A}\boldsymbol{v} \tag{13}$$

Both computations (density/likelihood and sampling) require $\mathbf{B}$ to be invertible and, therefore, that none of the variables have zero variance or are perfectly linearly-dependent on each other. Zero-variance variables must therefore be isolated and $\mathbf{B}$ marginalized before attempting to use Eq. (11) or to compute $\mathbf{A}$. Similarly, linear dependencies must also be identified beforehand. If we include variables one by one in the construction of $\mathbf{C}$, we can determine if a newly added one is linearly dependent if the determinant of the extended sample covariance matrix $\mathbf{C}$ is zero. Once identified, the regression coefficients for the dependent variable can be efficiently computed from $\mathbf{C}$ following the method described in [212]. The constant coefficient of the regression must also be calculated for future reference. What this process effectively does is to determine a linear model for each dependent variable that is represented by a set of regression coefficients. Dependent variables are not included in $\mathbf{C}$**,** but they need to be taken into account afterwards (e.g., by determining their values for the random samples by solving the linear model with the values obtained for the variables in $\mathbf{C}$).

5.2.3 High-Dimensional State Vectors

When the state vector of the model becomes large (i.e., d increases), as is the case for distributed high-resolution numerical models, difficulties start to arise when dealing with the computations involving the probability distribution. At first, the probability density, as computed with Eqs. (10) and (11), tends to diverge either towards zero or towards infinity. This phenomenon is related to the normalization of the density—so that it can integrate to one—and to its fast exponential decay as a function of the sample's distance from the kernel's centers. In these cases we propose replacing the density computation with an approximated likelihood formulation that is proportional to the inverse square Mahalanobis distance [213] to the root samples, thus skipping the exponentiation and normalization operations of the Gaussian density. This simplification, which corresponds to the inverse square difference between the sample value and the kernel's mean in the univariate case, is shown in Eq. (14). The resulting distortion of the Gaussian bell-curve shape does not affect the results significantly, given that OPTIMISTS uses the fitness functions only to check for domination between particles—so only the sign of the differences between likelihood values are important and not their actual magnitudes.

$$\mathcal{L}^{\text{Mahalanobis}}(\boldsymbol{s}|\boldsymbol{S}) = \frac{1}{\sum w_i} \sum_{i=1}^{n} \frac{w_i}{|(\boldsymbol{s} - \boldsymbol{s}_i)^{\mathrm{T}} \mathbf{B}^{-1} (\boldsymbol{s} - \boldsymbol{s}_i)|} \tag{14}$$

However, computational constraints might also make this simplified approach unfeasible both due to the $O(d^2)$ space requirements for storing the bandwidth matrix **B** and the $O(d^3)$ time complexity of the decomposition algorithms, which rapidly become huge burdens for the memory and the processors. Therefore, we can chose to sacrifice some accuracy by using a diagonal bandwidth matrix **B** which does not include any covariance term—only the variance terms in the diagonal are computed and stored. This implies that, even though the multiplicity of root samples would help in maintaining a large portion of the covariance, another portion is lost by preventing the kernels from reflecting the existing correlations. In other words, variables would not be rendered completely independent, but rather conditionally independent because the kernels are still centered on the set of root samples. Kernels using diagonal bandwidth matrices are referred to as "D-class" while those using the full covariance matrix are referred to as "F-class." The $k_{\mathrm{F-class}}$ parameter controls which version is used.

With only the diagonal terms of matrix **B** available (b_{jj}), we opt to roughly approximate the likelihood by computing the average of the standardized marginal likelihood value for each variable j, as shown in Eq. (15):

$$\mathcal{L}^{\text{independent}}(\boldsymbol{s}|\boldsymbol{S}) = \frac{1}{d\sqrt{2\pi}\sum w_i}\sum_{j=1}^{d}\sum_{i=1}^{n}\left\{w_i \cdot \exp\left[-\frac{\left(s_j - s_{i,j}\right)^2}{2b_{jj}}\right]\right\}, \tag{15}$$

where s_j represents the jth element of state vector $\boldsymbol{s}$, and $s_{i,j}$ represents the jth element of the ith sample of probability distribution $\boldsymbol{S}$. Independent/marginal random sampling of each variable can also be applied to replace Eq. (13) by adding random Gaussian residuals to the elements of the selected root sample $\boldsymbol{s}_{\text{root}}$.

5.3 Experimental Setup

In this section we prepare the elements to investigate whether if OPTIMISTS can help improve the forecasting skill of hydrologic models. More specifically, the experiments seek to answer the following questions: Which characteristics of Bayesian and variational methods are the most advantageous? How can OPTIMISTS be configured to take advantage of these characteristics? How does the algorithm compare to established data assimilation methods? And how does it perform with high-dimensional applications? To help answer these questions, this section first introduces two case studies and then it describes a traditional PF that was used for comparison purposes.

5.3.1 Case Studies

We coupled a Java implementation[2] of OPTIMISTS with two popular open-source distributed hydrologic modeling engines: Variable Infiltration Capacity (VIC) [8], [161]–[163], [214] and the Distributed Hydrology Soil and Vegetation Model (DHSVM) [10], [44]. Both engines needed several modifications so that they could be executed in a non-continuous fashion as required for sequential assimilation. Given the non-Markovian nature of surface routing schemes coupled with VIC that are based either on multiscale approaches [215], [216] or on the unit hydrograph concept [217], a simplified routing routine was developed that treats the model cells as channels—albeit with longer retention times. In the simplified method, direct runoff and baseflow produced by each model cell is partly routed through an assumed "equivalent" channel (slow component) and partly poured directly to the channel network (fast component). Both the channel network and the equivalent channels representing overland flow hydraulics are modelled using the Muskingum method. On the other hand, several important bugs in version 3.2.1 of the DHSVM, mostly related to the initialization of state variables but also pertaining to routing data and physics, were fixed.

We selected two watersheds to perform streamflow forecasting tests using OPTIMISTS: one with the VIC engine running at a 1/8th degree resolution for the Blue River in Oklahoma, and the other with the DHSVM running at a 100 m resolution for the Indiantown Run in Pennsylvania. Table 12 lists the main characteristics of the two test watersheds and the information of their associated model configurations. Figure 19 shows the land cover map together with the layout of

[2] Source code available at https://github.com/felherc/OPTIMISTS.

the modeling cells for the two watersheds. The multi-objective ensemble optimization algorithm introduced in Chapter 3 was employed to calibrate the parameters of the two models with the streamflow measurements from the corresponding USGS stations. For the Blue River, the traditional ℓ_2-norm Nash-Sutcliffe Efficiency (NSE_{ℓ_2}) (which focuses mostly on the peaks of hydrographs), an ℓ_1-norm version of the Nash-Sutcliffe Efficiency coefficient (NSE_{ℓ_1}) [218], and the mean absolute relative error MARE (which focuses mostly on the inter-peak periods) were used as optimization criteria. From 85,600 candidate parameterizations tried, one was chosen from the resulting Pareto front with NSE_{ℓ_2} = 0.69, NSE_{ℓ_1} = 0.56, and MARE = 44.71%. For the Indiantown Run, the NSE_{ℓ_2}, MARE, and absolute bias were optimized, resulting in a parameterization, out of 2,575, with NSE_{ℓ_2} = 0.81, MARE = 37.85%, and an absolute bias of 11.83 l/s.

These "optimal" parameter sets, together with additional sets produced in the optimization process were used to run the models and determine a set of time-lagged state variable vectors $\boldsymbol{s}$ to construct the state probability distribution $\boldsymbol{S}^0$ at the beginning of each of a set of data assimilation scenarios. The state variables include liquid and solid interception; ponding, water equivalent and temperature of the snow packs; and moisture and temperature of each of the soil layers. While we do not expect all of these variables to be identifiable and sensitive within the assimilation problem, we decided to be thorough in their inclusion—a decision that also increases the challenge for the algorithm in terms of the potential for overfitting. The Blue River model has 20 cells, with a maximum of seven intra-cell soil/vegetation partitions. After adding the stream network variables, the model has a total of d = 812 state variables. The Indiantown Run model has a total of 1,472 cells and d = 33,455 state variables.

Table 12 Characteristics of the Two Test Watersheds: Blue River and Indiantown Run

US hydrologic units are defined in [94]**. Elevation information was obtained from the Shuttle Radar Topography Mission** [95]**; land cover and impervious percentage from the National Land Cover Database** [67]**; soil type from CONUS-SOIL** [68]**; and precipitation, evapotranspiration, and temperature from NLDAS-2** [52]**. The streamflow and temperature include their range of variation of 90% of the time (5% tails at the high and low end are excluded).**

Model characteristic	**Blue River**	**Indiantown Run**
USGS station; US hydrologic unit	07332500; 11140102	01572950; 02050305
Area (km^2); impervious	3,031; 8.05%	14.78; 0.83%
Elevation range; average slope	158 m – 403 m; 3.5%	153 m – 412 m; 14.5%
Land cover	43% grassland, 28% forest, 21% pasture/hay	74.6% deciduous forest
Soil type	Clay loam (26.4%), clay (24.8%), sandy loam (20.26%)	Silt loam (51%), sandy loam (49%)
Avg. streamflow (90% range)	9.06 m^3/s (0.59 m^3/s – 44.71 m^3/s)	0.3 m^3/s (0.035 m^3/s – 0.793 m^3/s)
Avg. precipitation; avg. ET	1,086 mm/year; 748 mm/year	1,176 mm/year; 528 mm/year
Avg. temperature (90% range)	17.26°C (2.5°C – 31°C)	10.9°C (-3.5°C – 24°C)
Model cells; stream segments; d	20; 14; 812	1,472; 21; 33,455
Resolution	0.125°; daily	100 m; hourly
Calibration	167 parameters; 85 months; objectives: NSE_{ℓ_2}, NSE_{ℓ_1}, MARE	18 parameters; 20 months; objectives: NSE_{ℓ_2}, MARE, absolute bias

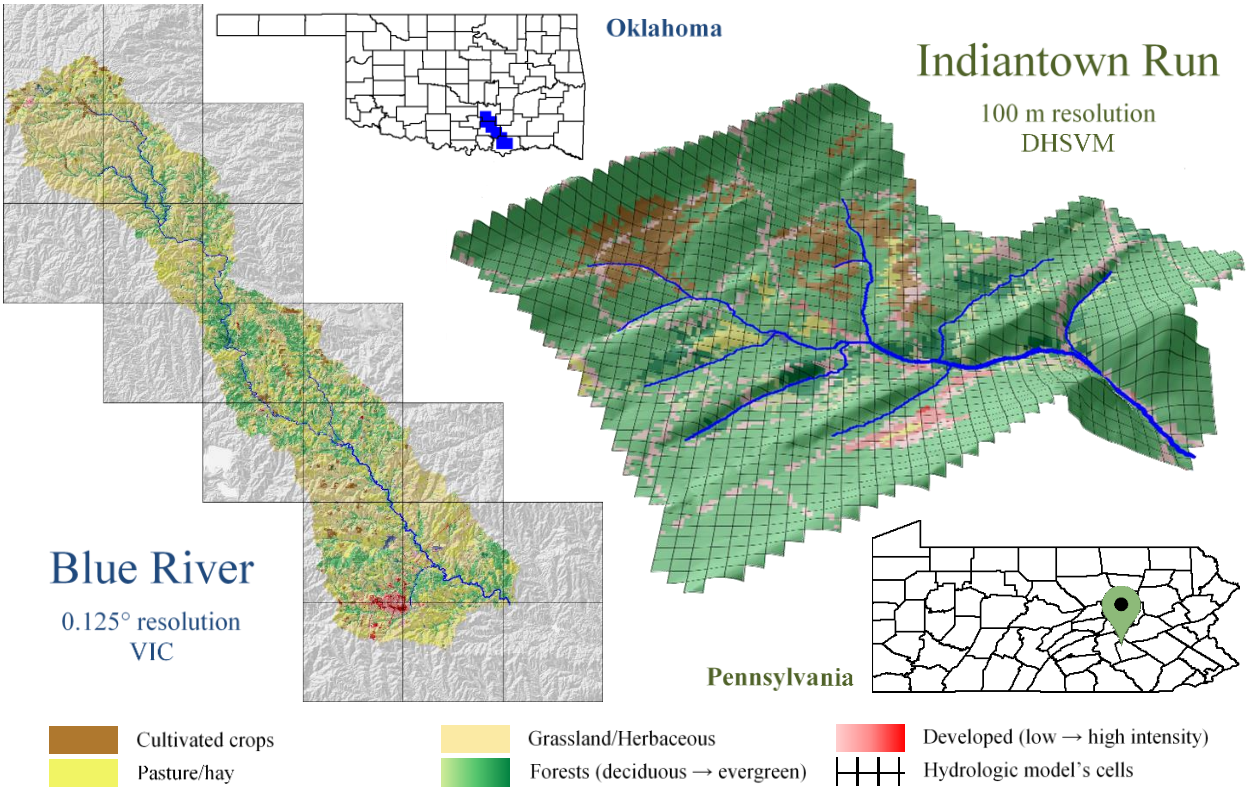


Figure 19 Maps of the Two Test Watersheds

The maps display the 30 m resolution land cover distribution from the NLCD [67]**. Left: Oklahoma's Blue River watershed 0.125° resolution VIC model (20 cells). Right: Pennsylvania's Indiantown Run watershed 100 m resolution DHSVM model (1,472 cells).**

Three diverse scenarios were selected for the Blue River, each of them comprised of a two-week assimilation period (when streamflow observations are assimilated), and a two-week forecasting period (when the model is run in an open loop using the states obtained at the end of the assimilation period): Scenario 1, starting on October 15th, 1996, is rainy through the entire four weeks. Scenario 2, which starts on January 15th, 1997, has a dry assimilation period and a mildly rainy forecast period. Scenario 3, starting on June 1st, 1997, has a relatively rainy assimilation period and a mostly-dry forecast period. Two scenarios, also spanning four weeks, were selected for the Indiantown Run, one starting on July 26th, 2009 and the other on August 26th, 2009.

We used factorial experiments [97] to test different configurations of OPTIMISTS on each of these scenarios, by first assimilating the streamflow and then measuring the forecasting skill. In this type of experimental designs a set of assignments is established for each parameter and then all possible assignment combinations are tried. The design allows to establish the statistical significance of altering several parameters simultaneously, providing an adequate framework for determining, for example, whether if using a short or a long assimilation time step Δt is preferable, or if utilizing the optional optimization step within the algorithm is worthwhile. Table 13 shows the setup of each of the three full factorial experiments we conducted, together with the selected set of assignments for OPTIMISTS' parameters. The forecasts were produced in an ensemble fashion, by running the models using each of the samples $\boldsymbol{s}_i$ from the state distribution $\boldsymbol{S}$ at the end of the assimilation time period, and then using the samples' weights w_i to produce an average forecast. Deterministic model parameters (those from the calibrated models) and forcings were used in all simulations.

Table 13 Setup of the Three Assimilation Factorial Experiments

The table includes the watershed, the total number of configurations (conf.), the values assigned to OPTIMISTS' parameters, and which objectives (objs.) were used (one objective: minimize MAE given the streamflow observations; two objectives: minimize MAE and maximize likelihood given source/background state distribution $\boldsymbol{S}^t$). n_{evo} was set to 25 in all cases. The total number of configurations results from combining all the possible parameter assignments listed for each experiment. Note that for Experiment 3 there are configurations that require a four-week assimilation period (all others have a length of two weeks).

No.	Watershed	Conf.	Δt	n	w_{root}	p_{samp}	$k_{\text{F-class}}$	g	objs.
1	Blue River	48	1d, 5d, 2w	100, 500	0.95	0.25, 1	false, true	0.75	1, 2
2	Indiantown Run	32	1h, 2w	100, 200	0.6, 0.95	0.25, 1	false	0.75	1, 2
3	Indiantown Run	24	1h, 6h, 1d, 3.5d, 2w, 4w	100	0.95	0.4, 1	false	0.5, 1	2

Observation errors are usually taken into account in traditional assimilation algorithms by assuming a probability distribution for the observations at each time step, and then performing a probabilistic evaluation of the predicted value of each particle/member against that distribution. As mentioned in Section 4.2 such a fitness metric, like the likelihood utilized in PFs to weight candidate particles, is perfectly compatible with OPTIMISTS. However, since it is difficult to estimate the magnitude of the observation error in general, and fitness metrics $\boldsymbol{f}_i$ here are only used to determine (non-)dominance between particles, we opted to use the mean absolute error (MAE) with respect to the streamflow observations in all cases.

For the Blue River scenarios, a secondary likelihood objective/metric was used in some cases to select for particles with higher consistency with the history. It was computed using either Eq. (15) if $k_{\mathrm{F-class}}$ was set to false, or Eq. (14) if it was set to true. Equation (15) was used for all Indiantown Run scenarios given the large number of dimensions. The assimilation period was of two weeks for most configurations, except for those in Experiment 3 which have Δt = 4 weeks. During both the assimilation and the forecasting periods we used unaltered streamflow data from the USGS and forcing data from NLDAS-2 [52]—even though a forecasted forcing would be used instead in an operational setting (e.g., from systems like NAM [6] or ECMWF [7]). While adopting perfect forcings for the forecast period leads to an overestimation of their accuracy, any comparisons with control runs or between methods are still valid as they all share the same benefit. Also, removing the uncertainty in the meteorological forcings allows the analysis to focus on the uncertainty corresponding to the land surface.

5.3.2 Data Assimilation Method Comparison

Comparing the performance of different configurations of OPTIMISTS can shed light into the adequacy of individual strategies utilized by traditional Bayesian and variational methods. For example, producing all particles with the optimization algorithms (p_{samp} = 0), setting long values for Δt, and utilizing a traditional two-term cost function as that in Eq. (6), makes the method behave somewhat as a hard-constrained 4D-Var approach; while sampling all particles from the source state distribution (p_{samp} = 1), setting Δt equal to the model time step, and using a single likelihood objective involving the observation error, would resemble a PF. Herein we also compare OPTIMISTS with a traditional PF on both model applications. Since the forcing is assumed to be deterministic, the implemented PF uses Gaussian “regularization”/perturbation of resampled particles to avoid degeneration [219]. Resampling is executed such that the probability of duplicating a particle is proportional to their weight [220].

Additionally, the comparison is performed using a continuous forecasting experiment setup instead of a scenario-based one. In this continuous test, forecasts are performed every time step and compiled in series for different forecast lead times that span several months. Forecast lead times are of 1, 3, 6, and 12 days for the Blue River and of 6 hours, and 1, 4, and 16 days for the Indiantown Run. Before each forecast, both OPTIMISTS and the PF assimilate streamflow observations for the assimilation time step of each algorithm (daily for the PF). The assimilation is performed cumulatively, meaning that the initial state distribution $\boldsymbol{S}^t$ was produced by assimilating all the records available since the beginning of the experiment until time t. The forecasted streamflow series are then compared to the actual measurements to evaluate their quality using deterministic metrics (NSE_{ℓ_2}, NSE_{ℓ_1}, and MARE) and two probabilistic ones: the ensemble-based continuous ranked probability score (CRPS) [221], which is computed for each time step and then averaged for the entire duration of the forecast; and the average normalized probability density p of the observed streamflow q_{obs} given the distribution of the forecasted ensemble $\boldsymbol{q}_{\mathrm{forecast}}$:

$$p(q_{\mathrm{obs}}|\boldsymbol{q}_{\mathrm{forecast}}) = \frac{\sum_{i=1}^{n} w_i \cdot \left(2\pi b^2\right)^{-2} \cdot \exp\left[-(q_{\mathrm{obs}} - q_i)^2/(2b^2)\right]}{\sum_{i=1}^{n} w_i}, \tag{16}$$

where the forecasted streamflow $\boldsymbol{q}_{\mathrm{forecast}}$ is composed of values q_i for each particle i and accompanying weight w_i, and b is the bandwidth of the univariate kernel density estimate. b can be obtained utilizing Silverman's rule of thumb [88]. The probability p is computed every time step and then averaged for all time steps. As opposed to the CRPS, which can only give an idea of the bias of the estimate, the density p can detect both bias and under/over-confidence: high values for the density indicate that the ensemble is producing narrow estimates around the true value, while low values indicate either that the stochastic estimate is spread too thin or is centered far away from the true value.

5.4 Results and Discussion

This section summarizes the forecasting results obtained from the three scenario-based experiments and the continuous forecasting experiments on the Blue River and the Indiantown Run models. We performed scenario-based experiments to explore the effects of multiple parameterizations of OPTIMISTS, and the performance was analyzed as follows. We ran the model for the duration of the forecast period (two weeks) using the state configuration encoded in each root state s_i of the distribution $\boldsymbol{S}$ obtained at the end of the assimilation period for each configuration of OPTIMISTS and each scenario. The mean streamflow time series was then computed for each case by averaging the model results for each particle $\boldsymbol{P}_i$ (the average was weighted based on the corresponding weights w_i). With this averaged streamflow series, we computed the three performance metrics—the NSE_{ℓ_2}, the NSE_{ℓ_1}, and the MARE—based on the observations from the corresponding stream gauge. With these, we computed the change in the forecast performance between each configuration and a control open-loop model run (one without the benefit of assimilating the observations).

5.4.1 Blue River – Low Resolution Model

Figure 20 summarizes the results for Experiment 1 with the model of the Blue River watershed using VIC, in which the distributions of the changes in MARE after marginalizing the results for each scenario and each of the parameter assignments are shown. That is, each box (and pair of whiskers) represents the distribution of change in MARE of all cases in the specified scenario or for which the specified parameter assignment was used. Negative values in the vertical axis indicate that OPTIMISTS decreased the error, while positive values indicate that it increased the error. It can be seen that, on average, OPTIMISTS improves the precision of the forecast in most cases, except for several of the configurations in Scenario 1 (for this scenario the control already produces a good forecast) and when using an assimilation step Δt of one day. We performed an analysis of variance (ANOVA) to determine the statistical significance of the difference found for each of the factors indicated in the horizontal axis. From the p-values for the main effects shown in Figure 20 we can conclude that the assimilation time step, the number of objectives, and the use of optimization algorithms are all statistically significant. On the other hand, the number of particles and the use of F-class kernels are not.

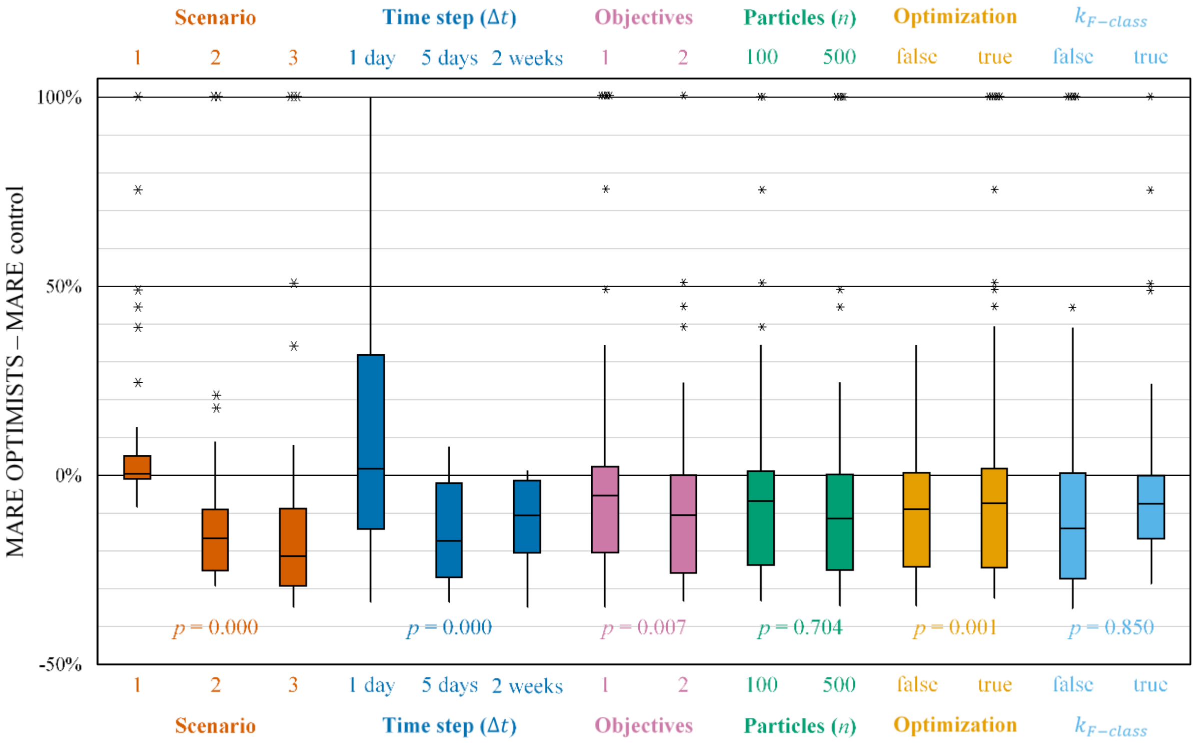


Figure 20 Summarized Results for Experiment 1 on the Blue River

Changes are relative to an open-loop control run where no assimilation was performed. Each column corresponds to the distribution of the error changes on the specified scenario or assignment to the indicated parameter. Positive values indicate that OPTIMISTS increased the error, while negative values indicate it decreased the error. Outliers are noted as asterisks and values were limited to 100%. For the one-objective case the particles' MAE was to be minimized; for the two-objective case, the likelihood given the background was to be maximized in addition. No optimization ("false") corresponds to p_{samp} = 1.0 (i.e., all samples are obtained from the prior distribution); "true" corresponds to p_{samp} = 0.25. The p-values were determined using ANOVA [97]**, and indicate the probability that the differences in means corresponding to boxes of the same color are produced by chance (e.g., values close to zero indicate certainty that the parameter effectively affects the forecast error).**

A Δt of five days produced the best results overall for the tested case, suggesting that there exists a sweet spot that balances the amount of information being assimilated (larger for a long Δt), and the number of state variables to be modified (larger for a small Δt). Based on such results, it is reasonable to assume that the sweet spot may depend on the time series of precipitation, the characteristics of the watershed, and the temporal and spatial resolutions of the model. From this perspective, the poor results for a step of one day could be explained in terms of overfitting, where there are many degrees of freedom and only one value being assimilated per step. Evaluating particles in the light of two objectives, one minimizing departures from the observations and the other maximizing the likelihood of the source state, resulted in statistically-significant improvements compared to using the first objective alone. Additionally, the data suggests that not executing the optional optimization step of the algorithm ("optimization = false"), but instead relying only on particles sampled from the prior/source distribution, is also beneficial. These two results reinforce the idea that maintaining consistency with the history to some extent is of paramount importance, perhaps to the point where the strategies used in Bayesian filters and variational methods are insufficient in isolation. Indeed, the best performance was observed only when both sampling was limited to generate particles from the prior state distribution and the particles were evaluated for their consistency with that distribution.

On the other hand, we found it counterintuitive that neither using a larger particle ensemble nor taking into account state-variable dependencies through the use of F-class kernels lead to improved results. In the first case it could be hypothesized that using too many particles could lead to overfitting, since there would be more chances of particles being generated that happen to match the observations better but for the "wrong reasons." In the second case, the non-parametric nature of kernel density estimation could be sufficient for encoding the raw dependencies between variables, especially in low-resolution cases like this one, in which significant correlations between variables in adjacent cells are not expected to be too high. Both results deserve further investigation, especially concerning the impact of D- vs. F-class kernels in high-dimensional models.

Interestingly, the ANOVA also yielded small p-values for several high-order interactions. This means that, unlike the general case for factorial experiments as characterized by the sparsity-of-effects principle [222], specific combinations of multiple parameters have a large effect on the forecasting skill of the model. Significant interactions (with p smaller than 0.05) are between the objectives and Δt (p = 0.001); n and $k_{\mathrm{F-class}}$ (p = 0.039); Δt and the use of optimization (p = 0.000); the use of optimization and $k_{\mathrm{F-class}}$ (p = 0.029); the objectives, Δt, and the use of optimization (p = 0.043); n, Δt, and $k_{\mathrm{F-class}}$ (p = 0.020); n, the use of optimization, and $k_{\mathrm{F-class}}$ (p = 0.013); and n, Δt, the use of optimizers, and $k_{\mathrm{F-class}}$ (p = 0.006). These interactions show that, for example, using a single objective is especially inadequate when the time step is of one day or when optimization is used. Also, employing optimization is only significantly detrimental when Δt is of one day—probably because of intensified overfitting, and that choosing F-class kernels leads to higher errors when Δt is small, n large, and the optimizers are being used.

Based on these results, we recommend the use of both objectives and no optimization as the preferred configuration of OPTIMISTS for the Blue River application. A time step of around five days appears to be adequate for this specific model. Also, without strong evidence for their advantages, we recommend using more particles or kernels of class F only if there is no pressure for computational frugality. However, the number of particles should not be too small to ensure an appropriate sample size.

Table 14 shows the results of the year-long continuous forecasting experiment on the Blue River using a 30-particle PF and a configuration of OPTIMISTS with a 7-day assimilation time step Δt, three objectives (NSE$_{\ell_2}$, MARE, and the likelihood), 30 particles, no optimization, and D-class kernels. This specific configuration of OPTIMISTS was chosen from a few that were tested that applied the recommendations above. The selected configuration was the one that best balanced the spread and the accuracy of the ensemble as some configurations had slightly better deterministic performance but larger ensemble spread for dry weather—which lead to worse probabilistic performance.

Table 14 Continuous Daily Forecast Performance Metrics for the Blue River

Comparison between OPTIMISTS (Δt = 7 days, 3 objectives: NSE$_{\ell_2}$, MARE, and likelihood; n = 30; no optimization; and D-class kernels) and a traditional PF (n = 30). The continuous forecast extends from January to June, 1997. The NSE$_{\ell 2}$, NSE$_{\ell 1}$, and MARE (deterministic) are computed using the mean streamflow of the forecast ensembles and contrasting it with the daily observations, while the CRPS and the density (probabilistic) are computed taking into account all the members of the forecasted ensemble.

Algorithm	Lead time	NSE$_{\ell 2}$	NSE$_{\ell 1}$	MARE	CRPS (m^3/s)	Density
OPTIMISTS	1 day	0.497	0.293	51.40%	7.173	0.061
	3 days	0.527	0.312	50.16%	6.959	0.065
	6 days	0.534	0.315	50.18%	6.945	0.073
	12 days	0.516	0.297	51.26%	7.124	0.078
Particle filter	1 day	0.675	0.522	30.06%	4.480	0.098
	3 days	0.623	0.493	33.20%	4.744	0.113
	6 days	0.602	0.473	35.79%	5.000	0.109
	12 days	0.515	0.432	38.36%	5.593	0.105

Both the selected configuration of OPTIMISTS and the PF methods show relatively good performance for all lead times (1, 3, 6, and 12 days) based on the performance metrics. However, the PF generally outperforms OPTIMISTS. Figure 21 shows the probabilistic streamflow forecasts for both algorithms for a lead time of 6 days. The portrayed evolution of the density, in which the mean does not necessarily correspond to the center of the ensemble spread, evidences the non-Gaussian nature of both estimates. We offer two possible explanations for this negative result. First, the relatively low dimensionality of this test case does not allow OPTIMISTS to showcase its real strength, perhaps especially since the large scale of the watershed does not allow for tight spatial interactions between state variables. Second, additional efforts might be needed to find a configuration of the algorithm, together with a set of objectives, that best suit the specific conditions of the tested watershed. While PFs remain easier to use “out of the box” because of their ease of configuration, the fact that adjusting the parameters of OPTIMISTS allowed to trade-off deterministic and probabilistic accuracy points to the adaptability potential of the algorithm, in which it allows for probing the spectrum between exploration and exploitation of candidate particles—which usually leads to higher and lower diversity of the ensemble, respectively.

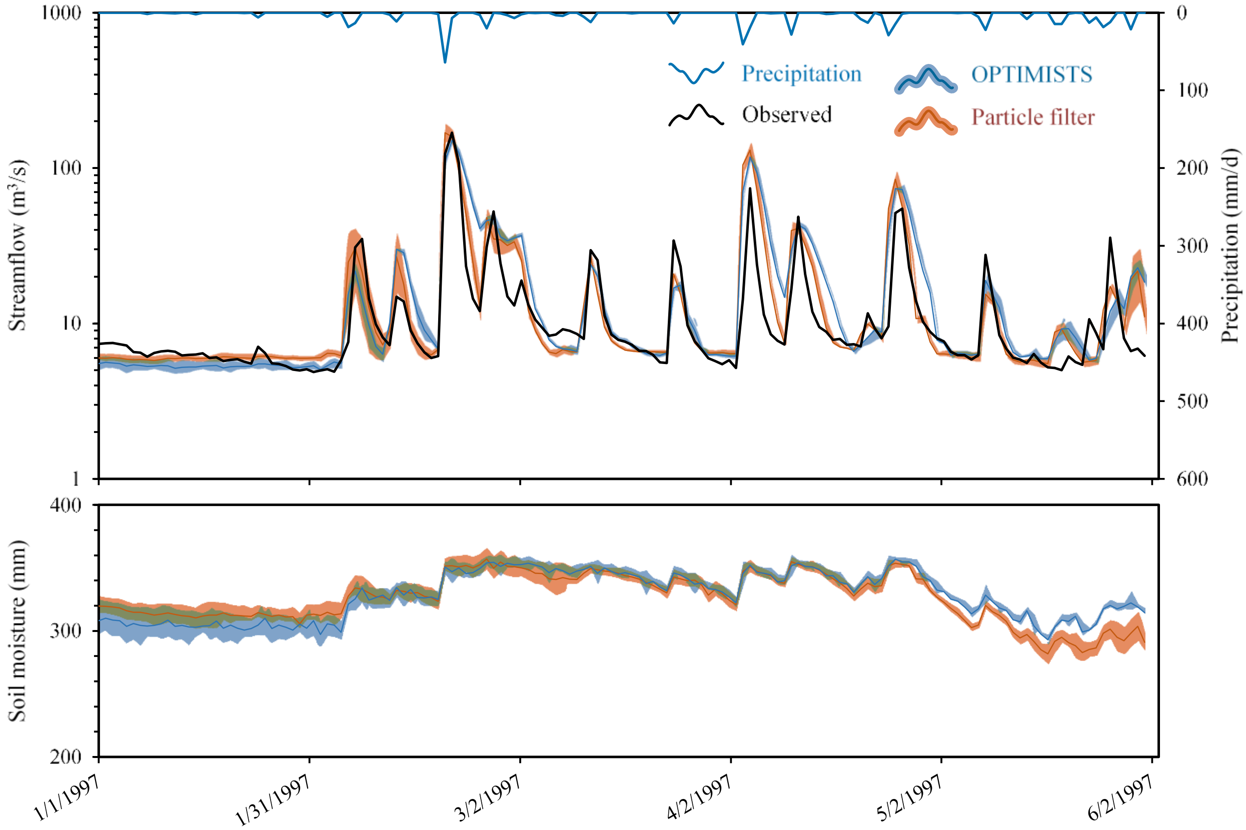


Figure 21 Comparison of Six-Day Lead Time Probabilistic Forecasts for the Blue River

Forecasts are of streamflow (top) and area-averaged soil moisture (bottom) using OPTIMISTS (Δt = 7 days; 3 objectives: NSE$_{\ell 2}$, MARE, and likelihood; n = 30; no optimization, and D-class kernels) and a traditional PF (n = 30). The dark blue and orange lines indicate the mean of OPTIMISTS' and the PF's ensembles respectively, while the light blue and light orange bands illustrate the spread of the forecast by highlighting the areas where the probability density of the estimate is at least 50% of the density at the mode (the maximum) at that time step. The green bands indicate areas where the light blue and light orange bands intersect.

5.4.2 Indiantown Run – High Resolution Model

Figure 22 summarizes the changes in performance when using OPTIMISTS in Experiment 2. In this case, the more uniform forcing and streamflow conditions of the two scenarios allowed to statistically analyze all three performance metrics. For Scenario 1 we can see that OPTIMISTS produces a general increase in the Nash-Sutcliffe coefficients, but a decline in the MARE, evidencing tension between fitting the peaks and the inter-peak periods simultaneously. For both scenarios there are configurations that performed very poorly, and we can look at the marginalized results in the boxplots for clues into which parameters might have caused this. Similar to the Blue River case, the use of a 1-hour time step significantly reduced the forecast skill, while the longer step almost always improved it; and the inclusion of the secondary history-consistency objective ("2 objectives") also resulted in improved performance. Not only does it seem that for this watershed the secondary objective mitigated the effects of overfitting, but it was interesting to note some configurations in which using it actually helped to achieve a better fit during the assimilation period.

While the ANOVA also provided evidence against the use of optimization algorithms, we are reluctant to instantly rule them out on the grounds that there were statistically significant interactions with other parameters (see the ANOVA table in the supplementary material). The optimizers led to poor results in cases with one-hour time steps or when only the first objective was used. Other statistically significant results point to the benefits of using the root samples more intensively (in opposition to using random samples) and, to a lesser extent, to the benefits of maintaining an ensemble of moderate size.

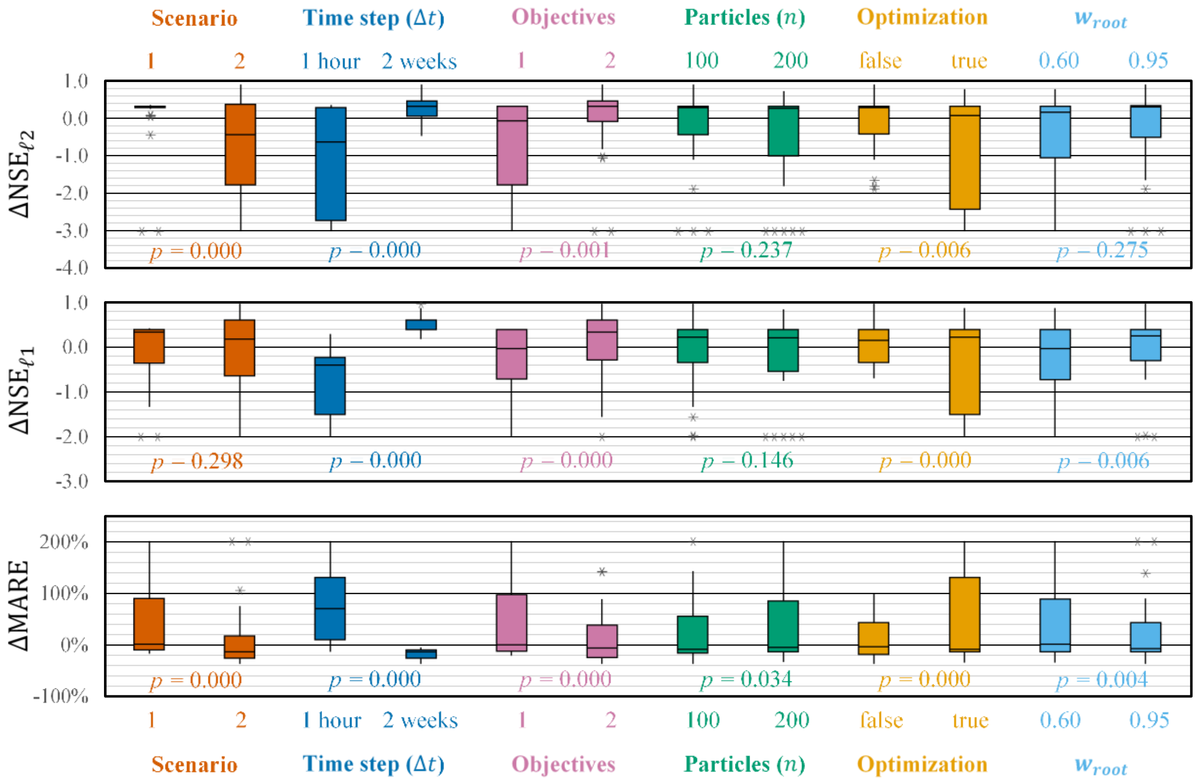


Figure 22 Summarized Results for Experiment 2 on the Indiantown Run

Changes are relative to an open-loop control run where no assimilation was performed. Each column corresponds to the distribution of the error metric changes on the specified scenario or assignment to the indicated parameter. Outliers are noted as stars and values were constrained to $\text{NSE}_{\ell 2} \geq -3$, $\text{NSE}_{\ell 1} \geq -3$, and MARE $\leq 200\%$. Positive values indicate improvements for the $\text{NSE}_{\ell 2}$ and the $\text{NSE}_{\ell 1}$. The meaning for the MARE and for other symbols are the same as those defined in Figure 20.

Figure 23 shows the summarized changes in Experiment 3, where the effect of the time step Δt is explored in greater detail. Once again, there appears to be evidence favouring the hypothesis that there exists a sweet spot, and in this case it appears to be close to the two weeks mark: both shorter and longer time steps led to considerably poorer performance. In this experiment, with all configurations using both optimization objectives, we can see that there are no clear disadvantages of using optimization algorithms (but also no advantages). Experiment 3 also shows that the effect of the greed parameter g is not very significant. That is, selecting some particles from dominated fronts to construct the target state distribution, and not only from the Pareto front, does not seem to affect the results.

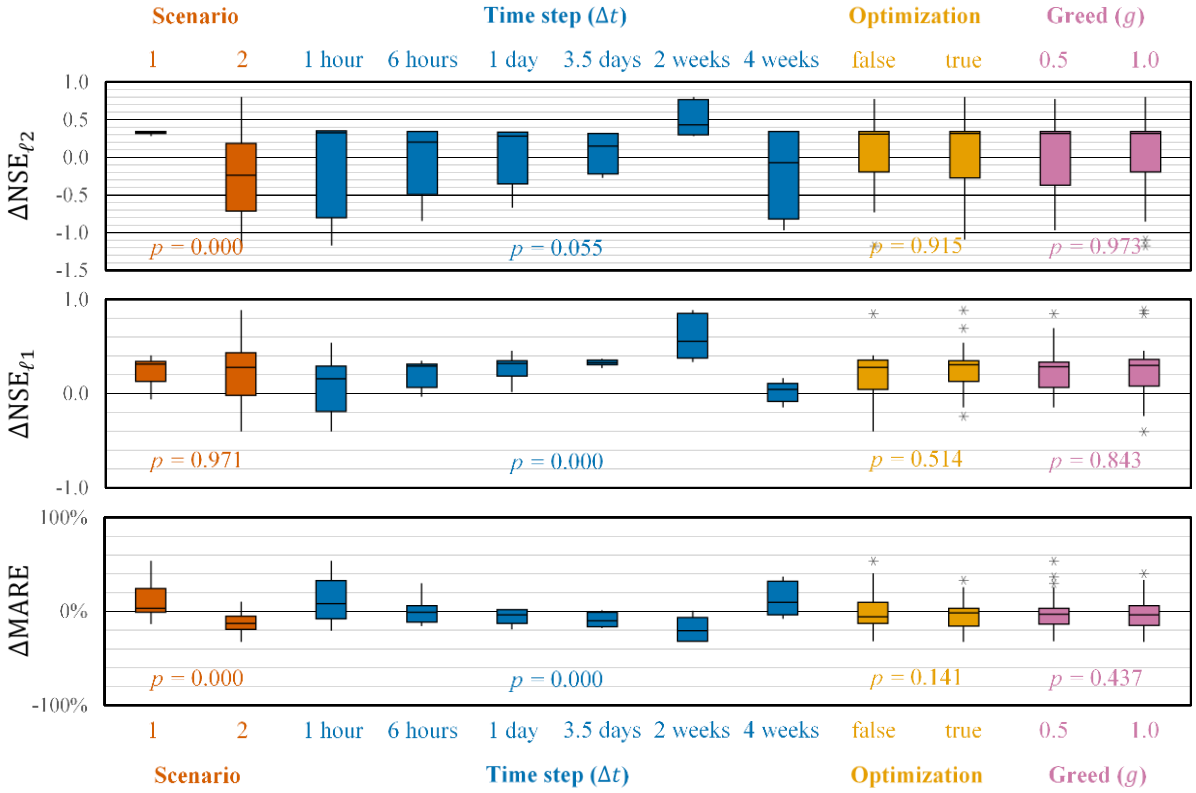


Figure 23 Summarized Results for Experiment 3 on the Indiantown Run

Changes are relative to an open-loop control run where no assimilation was performed. Each column corresponds to the distribution of the error metric changes on the specified scenario or assignment to the indicated parameter. Positive values indicate improvements for the $\mathrm{NSE}_{\ell 2}$ and the $\mathrm{NSE}_{\ell 1}$. See the caption of Figure 20 for more information.

Table 15 and Figure 24 show the results from comparing continuous forecasts from the PF and from a configuration of OPTIMISTS with a time step of one week, two objectives, 50 particles, and no optimization. Both algorithms display overconfidence in their estimations, which is evidenced in Figure 24 by the bias and narrowness of the ensembles' spread. It is possible that a more realistic incorporation of uncertainties pertaining to model parameters and forcings (which, as mentioned, are trivialized in these tests) would help compensate overconfidence. For the time being, these experiments help characterize the performance of OPTIMISTS in contrast with the PF, as both algorithms are deployed under the same circumstances. In this sense, while the forecasts obtained using the PF show slightly better results for lead times of 6 hours and 1 day, OPTIMISTS shows a better characterization of the ensemble's uncertainty for the longer lead times.

Table 15 Continuous Daily Forecast Performance Metrics for the Indiantown Run

Comparison between OPTIMISTS (Δt = 7 days, 2 objectives; n = 50; no optimization; and D-class kernels) and a traditional PF (n = 50). The continuous forecast extends from September to December, 2009. The $\mathrm{NSE}_{\ell 2}$, $\mathrm{NSE}_{\ell 1}$, and MARE (deterministic) are computed using the mean streamflow of the forecast ensembles and contrasting it with the daily observations, while the CRPS and the density (probabilistic) are computed taking into account all the members of the forecasted ensemble.

Algorithm	Lead time	$\mathrm{NSE}_{\ell 2}$	$\mathrm{NSE}_{\ell 1}$	MARE	CRPS (l/s)	Density
OPTIMISTS	6 hours	0.574	0.316	32.25%	97.27	0.016
	1 day	0.609	0.340	31.42%	93.92	0.013
	4 days	0.573	0.316	32.20%	97.19	0.025
	16 days	0.521	0.272	33.90%	103.51	0.013
Particle filter	6 hours	0.660	0.480	26.87%	79.61	0.061
	1 day	0.639	0.464	26.68%	82.75	0.051
	4 days	0.558	0.401	27.42%	93.20	0.021
	16 days	0.520	0.346	28.75%	102.37	0.010

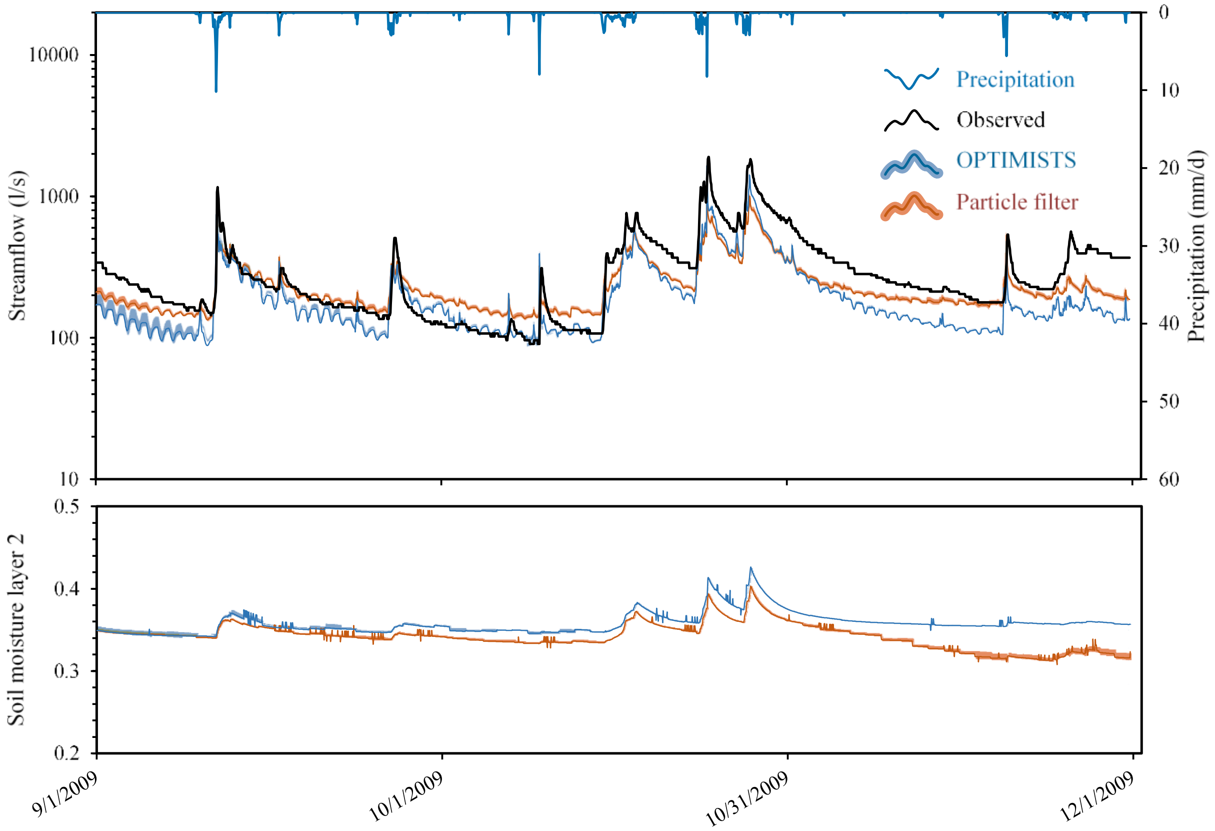


Figure 24 Comparison of 4-Day Lead Time Forecasts for the Indiantown Run

Forecasts are of streamflow (top) and area-averaged soil moisture (bottom) using OPTIMISTS (Δt = 7 days, 2 objectives, n = 50, no optimization, and D-class kernels) and a traditional PF (n = 50). The dark blue and orange lines indicate the mean of OPTIMISTS' and the PF's ensembles respectively, while the light blue and light orange bands illustrate the spread of the forecast by highlighting the areas where the probability density of the estimate is at least 50% of the density at the mode (the maximum) at that time step. The green bands indicate areas where the light blue and light orange bands intersect. Layer 2 of the soil corresponds to 100 to 250 mm depths.

OPTIMISTS' improved results in the high-resolution test case over those in the low-resolution one suggest that the strengths of the hybrid method might become more apparent as the dimensionality, and therefore the difficulty, of the assimilation problem increases. However, while OPTIMISTS was able to produce comparable results to those of the PF, it was not able to provide definite advantages in terms of accuracy. As suggested before, additional efforts might be needed to find the configurations of OPTIMISTS that better match the characteristics of the individual case studies. Moreover, the implemented version of the PF did not present the particle degeneracy or impoverishment problems usually associated with these filters when dealing with high dimensionality, which also prompts further investigation.

5.4.3 Computational Performance

It is worth noting that the longer the assimilation time step, the faster the entire process is. This occurs because, even though the number of hydrological calculations is the same in the end, for every assimilation time step the model files need to be generated accordingly, then accessed, and finally the result files written and accessed. This whole process takes a considerable amount of time. Therefore, everything else being constant, sequential assimilation (like with PFs) automatically imposes additional computational requirements. In our tests we used RAM drive software to accelerate the process of running the models sequentially and, even then, the overhead imposed by OPTIMISTS was consistently below 10% of the total computation time. Most of the computational effort remained with running the model, both for VIC and the DHSVM. In this sense, model developers may consider allowing their engines to be able to receive input data from main memory, if possible, to facilitate data assimilation and other similar processes.

5.4.4 Recommendations for Configuring OPTIMISTS

Finally, here we summarize the recommended choices for the parameters in OPTIMISTS based on the results of the experiments. In the first place, given their low observed effect, default values can be used for g (around 0.5). A w_{root} higher than 90% was found to be advantageous. The execution of the optimization step ($p_{\mathrm{samp}} < 1$) was, on the other hand, not found to be advantageous and, therefore, we consider it a cleaner approach to simply generate all samples from the initial distribution. Similarly, while not found to be disadvantageous, using diagonal bandwidth (D-class) kernels provide a significant improvement in computational efficiency and are thus recommended for the time being.

Even though only two objective functions were tested, one measuring the departures from the observations being assimilated and another measuring the compatibility of initial samples with the initial distribution, the results clearly show that it is beneficial to simultaneously evaluate candidate particles using both criteria. While traditional cost functions like the one in Eq. (6) do indeed consider both aspects, we argue that that using multiple objectives has the added benefit of enriching the diversity of the particle ensemble and, ultimately, the resulting probabilistic estimate of the target states.

Our results demonstrated that the assimilation time step is the most sensitive parameter and, therefore, its selection must be done with the greatest involvement. Taken the results together, we recommend that multiple choices be tried for any new case study looking to strike a balance between the amount of information being assimilated and the number of degrees of freedom. This empirical selection should also be performed with a rough sense of what is the range of forecasting lead-times that is considered the most important. Lastly, more work is required to provide guidelines to select the number of particles n to be used. While the literature suggests that more should increase forecast accuracy, our tests did not back this conclusion. We tentatively recommend trying different ensemble sizes based on the computational resources available and selecting the one that offers the best observed trade-off between accuracy and efficiency.

5.5 Conclusions and Future Work

In this chapter we introduced OPTIMISTS, a flexible, model-independent data assimilation algorithm that effectively combines the signature elements from Bayesian and variational methods: By employing essential features from particle filters, it allows performing probabilistic non-Gaussian estimates of state variables through the filtering of a set of particles drawn from a prior distribution to better match the available observations. Adding critical features from variational methods, OPTIMISTS grants its users the option of exploring the state space using optimization techniques and evaluating candidate states through a time window of arbitrary length. The algorithm fuses a multi-objective/Pareto analysis of candidate particles with kernel density probability distributions to effectively bridge the gap between the probabilistic and the variational perspectives. Moreover, the use of evolutionary optimization algorithms enables its efficient application on highly non-linear models as those usually found in most geosciences. This unique combination of features represent a clear differentiation from the existing hybrid assimilation methods in the literature [179], which are limited to Gaussian distributions and linear dynamics.

We conducted a set of hydrologic forecasting factorial experiments on two watersheds, the Blue River with 812 state variables and the Indiantown Run with 33,455, at two distinct modeling resolutions using two different modeling engines: VIC and the DHSVM, respectively. Capitalizing on the flexible configurations available for OPTIMISTS, these tests allowed to determine which individual characteristics of traditional algorithms prove to be the most advantageous for forecasting applications. For example, while there is a general consensus in the literature favoring extended time steps (4D) over sequential ones (1D-3D), the results from assimilating streamflow data in our experiments suggest that there is an ideal duration of the assimilation time step that is dependent on the case study under consideration, on the spatiotemporal resolution of the corresponding model application, and on the desired forecast length. Sequential time steps not only required considerably longer computational times but also produced the worst results—perhaps given the overwhelming number of degrees of freedom in contrast with the scarce observations available. Similarly, there was a drop in the performance of the forecast ensemble when the algorithm was set to use overly long time steps.

Procuring the consistency of candidate particles, not only with the observations but also with the history, led to significant gains in predictive skill. OPTIMISTS can be configured to both perform Bayesian sampling and find Pareto-optimal particles that trade-off deviations from the observations and from the prior conditions. This Bayesian/multi-objective formulation of the optimization problem was especially beneficial for the high-resolution test case, as it allows the model to overcome the risk of overfitting generated by the enlarged effect of equifinality.

On the other hand, our experiments did not produce enough evidence to recommend neither exploring the state space with optimization algorithms instead of doing so with simple probabilistic sampling, the use of a larger number of particles above the established baseline of 100, nor the computationally-intensive utilization of full covariance matrices to encode the dependencies between variables in the kernel-based state distributions. Nevertheless, strong interactions between several of these parameters suggest that some specific combinations could potentially yield strong outcomes. Together with OPTIMISTS' observed high level of sensitivity to the parameters, these results indicate that there could be promise in the implementation of self-adaptive strategies [223] to assist in their selection in the future. With these experiments, we were able to configure the algorithm to consistently improve the forecasting skill of the models compared to control open-loop runs. Additionally, comparative tests showed that OPTIMISTS was able to reliably produce adequate forecasts that were comparable to those resulting from assimilating the observations with a particle filter in the high-resolution application. While not being able to provide consistent accuracy advantages over the implemented particle filter, OPTIMISTS does offer considerable gains in computational efficiency given its ability to analyze multiple model time steps each time.

Moreover, in this article we offered several alternatives in the implementation of the components of OPTIMISTS whenever there were tensions between prediction accuracy and computational efficiency. In the future, we will focus on incorporating additional successful ideas from diverse assimilation algorithms and on improving components in such a way that both of these goals are attained with ever-smaller compromises. For instance, the estimation of initial states should not be overburdened with the responsibility of compensating structural and calibration deficiencies in the model. In this sense, we embrace the vision of a unified framework for the joint probabilistic estimation of structures, parameters, and state variables [177], where it is important to address challenges associated with approaches that would increase the indeterminacy of the problem by adding unknowns without providing additional information or additional means of relating existing variables. We expect that with continued efforts OPTIMISTS will be a worthy candidate framework to be deployed in operational settings for hydrologic prediction and beyond.

6.0 High-Resolution Model Initialization I: Compression of State Variables

This chapter presents a strategy for reducing the dimensionality of state variable representations to help in the process of data assimilation. The chapter is based on the article "Efficient data assimilation in high-dimensional hydrologic modeling through optimal spatial clustering" by Felipe Hernández and Xu Liang, submitted to the 2019 Environmental and Water Resources Institute (EWRI) conference to be held in Pittsburgh in May 19-23. The article, to be published on the conference proceedings, is currently in press.

6.1 Introduction

Many data assimilation algorithms represent the spatial relationships and dependencies between state variables through a covariance matrix. However, as the detail and/or extent of the model increase, and thus so the number of state variables n, the memory requirements for storing the matrix grow quadratically (n^2) and the number of operations required to extract information grows cubically (n^3). This dramatically rises the requirements of computational resources to perform forecasts with complex models.

One possible way to simplify representations of relationships between variables is by focusing only on the important ones. That is, if two variables are loosely related, perhaps because they correspond to far-apart locations, it can be assumed that they are, in practice, independent from each other. This principle may be called "distance decoupling," and is at the heart of several methods in the literature. For example, banded covariance matrices, which only have non-zero values in a band centered on the main diagonal, have helped meteorological and oceanographic applications for years [176]. More recent methods have relied on mathematical techniques to reduce the rank of the covariance through eigenvalue decomposition [224], or to operate over related matrices that allow for faster computations [225], [226].

On the other hand, there are two unexplored paths in the quest for efficient data assimilation. The first one runs in opposition to the distance decoupling principle: instead of abandoning weak relationships, it fully embraces the strongest ones. In other words, tightly related variables (e.g., because of spatial adjacency) can be assumed to change in a fully-coupled fashion. This principle may be called "proximity coupling." The second path is the use of physical principles to define efficient interaction representations. That is, to use domain-specific knowledge when prioritizing which interactions to incorporate into the analysis, and in which manner. These two paths have met before in the development of hydrologic models that seek to reduce the spatial resolution of watershed representations such that the least possible amount of information is lost. For instance, SMART aggregates model cells into equivalent cross-sections that can be solved linearly instead of spatially [227] and, similarly, HydroBlocks uses k-means clustering [228] to group cells with similar expected behavior together [229].

These types of compressed-resolution models could be used for forecasting applications and benefit from low-cost data assimilation. However, benefits from compression techniques can also be achieved during assimilation with fully distributed hydrologic models. For this purpose, this chapter introduces a clustering approach that seeks to group together state variables of similar cells in highly distributed hydrologic models—in an effort to accelerate the process of data assimilation. The Methods section first describes how to optimally determine which cells in a model should be clustered together, and then how can this technique be incorporated into an existing assimilation algorithm. This study used the DHSVM and OPTIMISTS. The Experimental setup section describes the watershed and the set of runs used to determine the advantages and/or disadvantages of utilizing the clustering method, and whether if there exists a preferred level of compression. The remainder two sections analyze the results and provide recommendations for practical implementations of the algorithm and for future research.

6.2 Methods

This section describes the proposed approach to make hydrologic data assimilation more efficient by reducing the size of the vector of state variables.

6.2.1 State Compression

This subsection presents the method for compressing and decompressing the vector of state variables $\boldsymbol{s}$. To simplify the explanation, it is assumed that only state variables in soil column elements (i.e., the sub-watersheds in semi-distributed models, or the pixels/cells in distributed/gridded ones) are included in $\boldsymbol{s}$. The level of similarity between elements is determined using one or more scalar distance functions d that operate over the state variables of two different cells of the model $\boldsymbol{s}^a$, $\boldsymbol{s}^b$. A possible form of the function d is a weighted Euclidean distance of the state variables in each cell with a weight vector $\boldsymbol{w}$:

$$d_{\text{Euc}}(\boldsymbol{s}^a, \boldsymbol{s}^b) = \sqrt{\frac{\sum_j \left[{w_j}^2 \cdot \left(s_j^b - s_j^a \right)^2 \right]}{\sum_j {w_j}^2}} \tag{17}$$

Given a set of reference states $\boldsymbol{s}^c$, a grouping (or clustering) function $\boldsymbol{g}$ can be used to compute a reduced representation of the state variables of a hydrologic model ${}^{-}\boldsymbol{s}$ from the original vector of state variables $\boldsymbol{s}$:

$${}^{-}\boldsymbol{s} = \boldsymbol{g}(\boldsymbol{s}, d, \{\boldsymbol{s}^c \; \forall \; c\}) \tag{18}$$

This is done by determining, for each cell l, which reference state $\boldsymbol{s}^c$ is the closest according to d:

$$\alpha_l = \underset{\boldsymbol{c}}{\operatorname{argmin}} \; d(\boldsymbol{s}^l, \boldsymbol{s}^c) \tag{19}$$

After determining all assignments α_l, a centroid $\overline{\boldsymbol{s}^c}$ is computed for each group (also known as cluster or super pixel) c as the average of all the cell vectors $\boldsymbol{s}^l$ in the cluster:

$$\overline{s_j^c} = \frac{\sum_j \beta_{l,c} \, s_j^l}{\sum_j \beta_{l,c}}, \beta_{l,c} = \begin{cases} 1 & \text{if } \alpha_l = c \text{ and } s_j \in \boldsymbol{s}^l \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

The compressed state vector ${}^{-}\boldsymbol{s}$ is created as the ordered concatenation of the centroids $\overline{\boldsymbol{s}^c}$. This study uses the traditional k-means algorithm [230] to determine the reference states $\boldsymbol{s}^c$, although any other clustering method could be used instead. The k-means algorithm has the following steps:

1. (Randomly) initialize k centroids $\boldsymbol{s}^c$.
2. Compute assignments α_l for every cell l using Eq. (19).
3. Update the centroids $\boldsymbol{s}^c$ as the average of all cells currently assigned to them.
4. If no changes were made to any of the centroids $\boldsymbol{s}^c$, return; otherwise, go back to 2.

Decompression, the process by which $\bar{\boldsymbol{s}}$ is re-projected to a full representation $\boldsymbol{s}'$ (with values for all model elements) can be performed by assigning the values of each centroid $\boldsymbol{s}^c$ to all the cells l that make part of each cluster c:

$$\boldsymbol{s}' = \{\boldsymbol{s}'^l \; \forall \; l \in \boldsymbol{\mathcal{M}}\} \tag{21}$$

$$\boldsymbol{s}'^l = \boldsymbol{s}^c|_{c=\alpha_l} \tag{22}$$

This can be expressed by notating the transformation through an inverse $\boldsymbol{g}$ function $\boldsymbol{g}^{-1}$:

$$\boldsymbol{s}' = \boldsymbol{g}^{-1}(\bar{\boldsymbol{s}}, d, \boldsymbol{s}^c) \tag{23}$$

Moreover, smooth state representations can be produced when transitioning from the reduced vector $\bar{\boldsymbol{s}}$ back to the original one by using principles from "fuzzy" clustering [231], in which the hard group assignments of α_l are softened. For example, this can be achieved through interpolation using inverse square distances to all centroids:

$$s'^l_i = \frac{\sum_c \frac{s_i^c}{[d(\boldsymbol{s}^l, \boldsymbol{s}^c)]^2}}{\sum_c \frac{1}{[d(\boldsymbol{s}^l, \boldsymbol{s}^c)]^2}} \tag{24}$$

Finally, with the ability to compress and decompress, the optimal distance function d^* can be found as the distance function d which minimizes an error metric e that compares the results of running the model $\boldsymbol{\mathcal{M}}$ with state vectors $\boldsymbol{s}$ and $\boldsymbol{s}'$:

$$d^* = \underset{d}{\operatorname{argmin}} \; e[\boldsymbol{\mathcal{M}}(\boldsymbol{p}, \boldsymbol{s}, \boldsymbol{f}), \boldsymbol{\mathcal{M}}(\boldsymbol{p}, \boldsymbol{s}', \boldsymbol{f})] \tag{25}$$

If using the Euclidean distance d_{Euc}, the problem becomes finding the optimal values of the variable weight vector $\boldsymbol{w}$ with $\boldsymbol{s}' = \boldsymbol{g}^{-1}[\boldsymbol{g}(\boldsymbol{s}, d_{\text{Euc}}, \boldsymbol{s}^c), d_{\text{Euc}}, \boldsymbol{s}^c]$:

$$d^*_{\text{Euc}} = \underset{\boldsymbol{w}}{\operatorname{argmin}} \; e[\boldsymbol{\mathcal{M}}(\boldsymbol{p}, \boldsymbol{s}, \boldsymbol{f}), \boldsymbol{\mathcal{M}}(\boldsymbol{p}, \boldsymbol{s}', \boldsymbol{f})] \tag{26}$$

6.2.2 Integration with Data Assimilation

The proposed method was incorporated into OPTIMISTS. The compressed-state version of OPTIMISTS modifies the sampling, simulation, and evaluation steps. In the sampling step, the centroids $\boldsymbol{s}^c$ of k clusters are defined using the k-means algorithm specified in the previous sub-section. The inputs are all of the cell state variable vectors $\boldsymbol{s}_i^l$ for each cell l in each state vector $\boldsymbol{s}_i$ in $\boldsymbol{S}^t$. That is, the distribution $\boldsymbol{S}^t$ is first divided into all its sample vectors $\boldsymbol{s}_i$, and then each is divided into its component cells/pixels $\boldsymbol{s}_i^l$. For higher computational efficiency, only a fraction of the states $\boldsymbol{s}_i$ (preferably those with higher weight) can be used to determine the centroids $\boldsymbol{s}^c$. Once the centroids $\boldsymbol{s}^c$ are available, the compressed state vectors $\bar{\boldsymbol{s}}$ are determined for the root samples using Eq. (18). The compressed samples $\bar{\boldsymbol{s}}$ are then used to generate a kernel density distribution $\bar{\boldsymbol{S}}$ from which random samples are generated.

In the simulation step, the compressed samples $\bar{\boldsymbol{s}}$ that were generated randomly need to be decompressed ($\boldsymbol{s}'$) in order to run their corresponding hydrologic simulation using the model. This is accomplished through the use of the Eq. (23) and Eq. (24). Both the compressed $\bar{\boldsymbol{s}}$ and decompressed $\boldsymbol{s}'$ state vectors are saved to be used in the latter steps. In the evaluation step, it is recommended to judge candidate samples based, not only on the corresponding discrepancies with the observations but, on the consistency with the state variable history—what is referred to in variational assimilation as minimizing the background error [195]. To this effect, the compressed state $\bar{\boldsymbol{S}}$ can be used to compute the likelihood of candidate samples more efficiently than the full distribution $\boldsymbol{S}$.

These modifications come at the additional cost of including the execution of the k-means algorithm, and the compression and decompression computations. However, with them, the compressed-state version of OPTIMISTS has two potential advantages that may make the cost justified. First, the reduced dimensionality of the state vectors (from $\boldsymbol{s}$ to $\bar{\boldsymbol{s}}$) allows for increased tractability in the matrix computations that are usually utilized for sampling the state variable distributions and for computing the state history consistency [168]. Second, even if the matrix operations are still out of reach (perhaps because of the limited number of samples available), and independence needs to be assumed between state variables, the proposed clustered approach guarantees a level of coupling between variables (the proximity coupling) that may preserve meaningful physical spatial interactions that would be lost otherwise.

Finally, it should also be noted that, unlike proximity coupling approaches previously used to create hydrologic models [227], [229], the clustering method proposed here is dynamic in the sense that groupings are constantly modified at each assimilation time step Δt following changes in the spatial distribution of state variables $\boldsymbol{s}$.

6.3 Experimental Setup

This section introduces the experimental setup utilized to test whether if the proposed compressed version of OPTIMISTS can improve the accuracy of forecasts with high resolution models, and what level of compression is better suited for this task. For this purpose the DHSVM Indiantown Run model is used. The data assimilation experiment required first determining the parameters of the k-means compression algorithm, including the number of clusters k and the weights w_i for each of the variables involved in the distance function d_{Euc}. Three distinct distance functions were used in the compression process, one for the clustering of interception variables, one for the snow cover variables, and one for the soil moisture variables. To achieve this, variables on each of these categories were clustered separately (by dividing the vectors s^l into three groups) and then the values at the centroids of each clustering $\bar{\boldsymbol{s}}^c$ were concatenated in the final compressed state ${}^{-}\boldsymbol{s}$. This results in instances of cells having interception-related or snow cover-related variables that belong to a different cluster than its soil moisture-related ones.

The optimization problem posed in Eq. (26) was used to find the optimal weights for each set of variables, together with the optimal number of clusters. For this purpose, the model simulated streamflow time series for eight one-week scenarios (8/18/08, 11/22/08, 12/1/08, 12/8/08, 1/7/09, 7/26/09, 8/26/09, and 11/17/09) that spanned a broad range of weather conditions. The multi-objective evolutionary ensemble optimization algorithm described in Chapter 2 helped to find the compression configurations that minimized the streamflow mean absolute error in these simulations. The number of clusters was set as a second objective value to be minimized in order to avoid the population converging to the maximum number of clusters (i.e., no compression).

A factorial experiment [97] was designed to determine the statistical significance of the effect of the level of compression applied to the state variables. Five compression levels were determined from the weight optimization process: “no,” “lower,” “low,” “high,” and “higher.” Moreover, given the importance of the assimilation time step Δt and its dependence on the specific model [168], [194], the experiment also included three different alternatives: 6 hours, 2 days, and 1 week. Additional OPTIMISTS parameters were set as follows: 50 particles, diagonal covariance, no optimization, 95% root samples, and greed = 0.75. Each run in the experiment consisted of performing continuous forecasts for a set of lead times (6 hours, 1 day, 2 days, and 16 days) and comparing the predicted streamflow for these lead times with the actual observations from the USGS station. For this purpose, data was assimilated within a moving window for the defined time step Δt, and then the forecast was executed for 16 days, noting the corresponding predicted values at the evaluation lead times. The window was moved to span a three-month time period between 9/1/2009 and 12/1/2009, which included some unusually-heavy precipitation events.

Three performance metrics evaluated the resulting ensemble forecasts in contrast with the observations: the NSE of the ensemble mean, the continuous ranked probability score (CRPS) [221], and the average probability density p of the observed streamflow q_{obs} given the distribution of the forecasted ensemble $\boldsymbol{q}_{\text{forecast}}$ (Eq. (16)). The density values p for the forecasted time series are averaged to yield the final density score. The NSE is a deterministic metric, as it only measures the distance between the ensemble mean and the observations. The CRPS improves the NSE by providing the average error of each of the ensemble members. The density provides the best probabilistic interpretation of the performance, as it gives a sense, not only of the error of the members but also, of under- or over-confidence. The results from the 60 runs in the experiment (5 compression levels $\times$ 3 time steps Δt $\times$ 4 lead times) were subjected to an analysis of variance (ANOVA) to determine the statistical significance of the individual factors and their interactions [97] on the performance. The source code of the method, together with the scripts to run the weight optimization process and the assimilation experiment in this article are available in OPTIMISTS's GitHub repository.

6.4 Results and Discussion

The cluster number k and variable weight $\boldsymbol{w}$ optimization process evaluated a total of 29,563 candidate configurations. Given that each configuration was judged in a multi-objective fashion, the final population contains a group of them that represent the trade-offs between decreasing the discrepancies with the control simulation (where no compression is applied), and increasing the level of compression. The fitness values of this resulting Pareto front are shown in Figure 25. For reference, the average streamflow in the Run is around 300 l/s and the number of state variables in the uncompressed state is 33,455. As expected, higher levels of compression led to larger errors. The error-vs-compression curve seems to portray a power law, in which the error decreases fast when compression is high, but sees only marginal improvements with smaller compression rates.

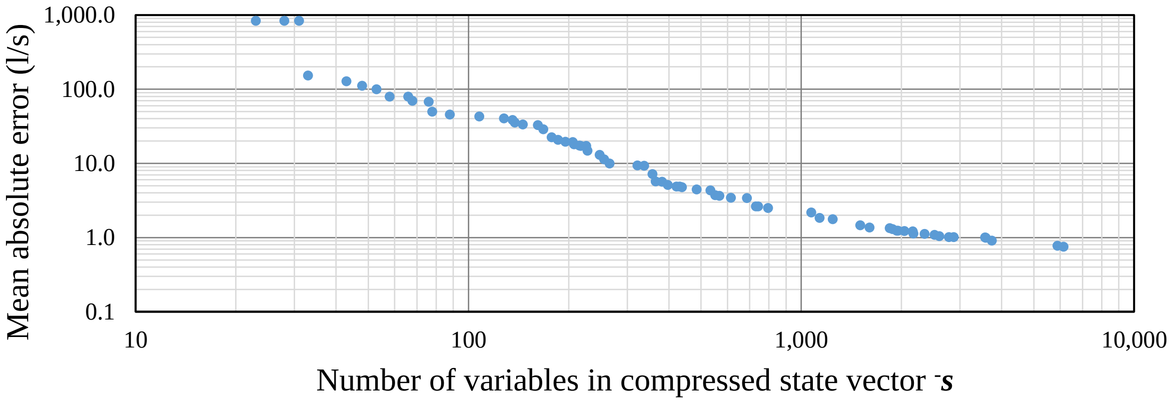


Figure 25 Model Error After Clustering-Declustering Cycle

Obtained as the final Pareto front of a cluster count and weight optimization process.

Four compression levels, to be used for the assimilation experiment, were taken from these results. Table 16 shows, for each of these chosen configurations, the values for k for each of the three clustering distances d_{Euc}, and the total number of variables in the compressed state $\bar{\boldsymbol{s}}$. The main factorial experiment was conducted using these four compression options and also included a control configuration with no compression. Table 16 also shows the lower bound of the percentage of time used in each case to run the model, as opposed to running the other steps in

OPTIMISTS. In general, lower compression requires more computations, as the number of clusters into which the model cells are sorted increases. These costs could potentially be offset if a full covariance matrix is used instead of a diagonal one to characterize inter-variable dependencies.

Table 16 Characteristics of Compression Configurations

The table shows the total number of variables in the compressed state vector ^{-}s, the compression percentage, number of clusters k, and the percentage of assimilation time dedicated to running the model for each compression level.

Parameter/weight (units)	**Compression level**				
	No	**Lower**	**Low**	**High**	**Higher**
Total number of variables	33,455	2,194	814	229	64
Compression percentage	0.0%	93.4%	97.6%	99.3%	99.8%
Number of clusters k (interception)	-	5	7	2	1
Number of clusters k (snow)	-	52	7	1	1
Number of clusters k (soil)	-	174	71	20	4
Modeling time ($\Delta t = 6$ hours)	92.5%	22.8%	51.9%	61.3%	81.3%
Modeling time ($\Delta t = 2$ days)	91.2%	24.3%	41.9%	78.1%	83.2%
Modeling time ($\Delta t = 1$ week)	91.1%	42.4%	64.2%	78.9%	89.2%

An analysis of variance (ANOVA) was used to determine the statistical significance of utilizing different levels of compression and different time steps Δt. Table 17 shows the p-values from the ANOVA. The results thus show that both the compression level and the time step Δt affect all the forecast performance metrics; except for the NSE, in which only Δt was found to have a significant effect. The interaction (compression $\times$ Δt) of these two factors also has a significant effect on the CRPS and, to a lower degree, on the density. This means that the choice of the compression level and the time step need to be taken into account in order to produce more accurate forecasts.

Table 17 P-values from the ANOVA for the Three Forecast Performance Metrics

Values close to zero indicate that the listed factor has a statistically significant effect on the assimilation (DA) performance.

Source	**P-value**		
	NSE	**CRPS**	**Density**
Model	0.000	0.000	0.000
Lead time	0.000	0.000	0.138
Linear terms	0.000	0.000	0.000
Compression	0.969	0.000	0.001
DA time step Δt	0.000	0.000	0.000
Compression × Δt	0.999	0.000	0.039

Figure 26 illustrates which compression and Δt choices led to better forecasts, by showing the range of scores of configuration groups. For the NSE, the uniform behavior across compression levels is in contrast with the clear advantage of using the shorter Δt. Conversely, a choice of a low compression level and a Δt of two days seems to be advantageous in terms of the CRPS and, especially, the density. This suggests that there should exist an intermediate level of compression in which the probabilistic accuracy in maximized. Similarly, the existence of an intermediate Δt with higher performance is consistent with previous findings [168]. In practice, it should be found individually for the specific configurations of OPTIMISTS used and for the specific model.

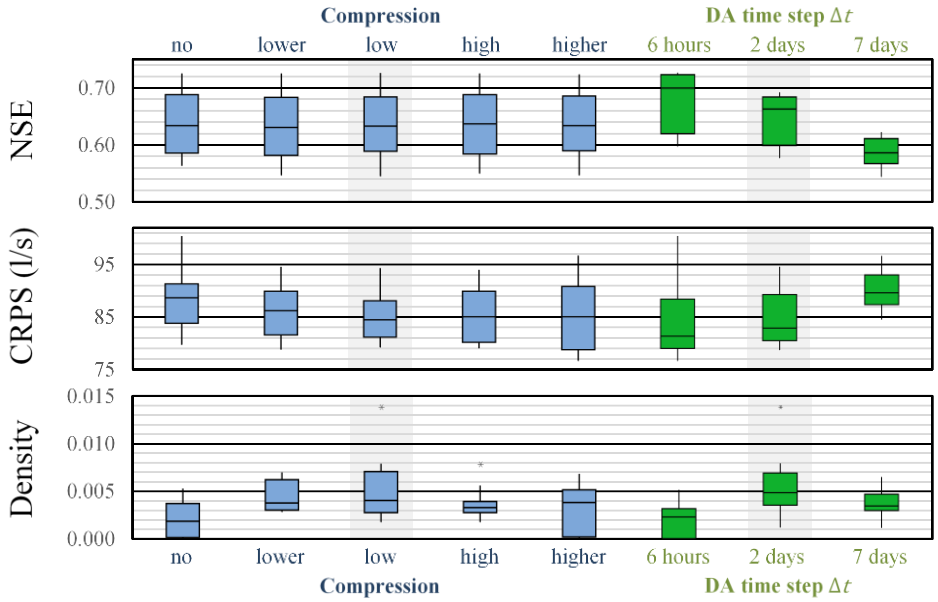


Figure 26 Boxplots of the Three Forecast Performance Metrics

Each column corresponds to the set of configurations with either a specific level of compression or with a specific time step Δt. Higher is better for the NSE and the density; lower is better for the CRPS.

As an example of the resulting forecasts, Figure 27 shows the estimated 2 days lead time average hydrographs for the three-month window for two of the 60 runs in the experiment: no compression, $\Delta t = 2$ days; and low compression, $\Delta t = 2$ days (highlighted configuration in Figure 26). While both forecasts have similar biases, the superior density score suggests that using compression helps in reducing the harmful effects of overconfidence to a certain degree. That is, estimates with uncompressed states result in very low variance/uncertainty despite their bias and, in some cases, the entire ensemble collapses to a single deterministic estimate. Compression, through the coupling of related variables, appears to allow for a better representation of the uncertainties in the state variable.

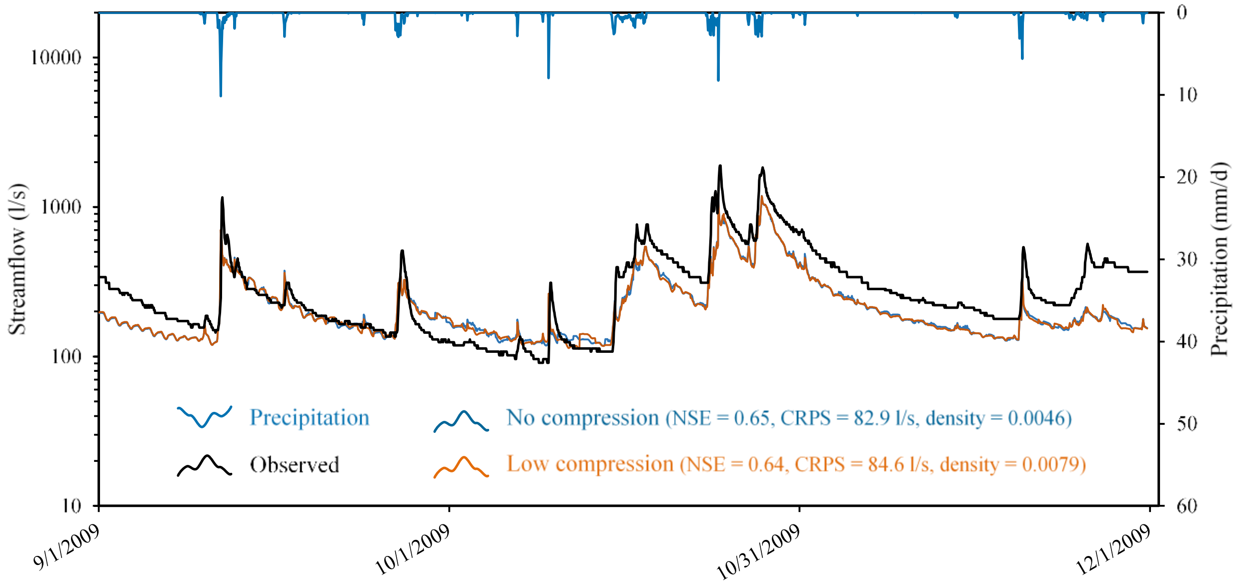


Figure 27 Comparison of 4-day Lead Time Streamflow Forecasts

Comparison between a no-compression configuration, a low compression configuration, and the actual observations.

6.5 Conclusions and Future Work

This chapter introduced an algorithmic extension to common data assimilation methods that allows to efficiently represent high-resolution model states without having to operate on full covariance matrices. The proposed technique achieves this by coupling state variables that behave similarly using fine-tuned instances of the classical k-means algorithm. As opposed to other methods proposed in the literature, in which "distant" variables are fully decoupled, efficiency is gained here by allowing grouped/clustered variables to change together as part of a "super pixel." The physically-based method utilizes one or more sets of variables to be clustered according to optimized distance functions. Additionally, a fuzzy decompression approach allows the posterior recovery of full state representations that retain an adequate level of spatial smoothness. The distance functions can be calibrated by minimizing the change in performance of simulations run before and after undergoing the compression and decompression processes.

The proposed method was incorporated into OPTIMISTS. The method was tested for changes in accuracy on a continuous hydrologic forecasting experiment on a high-resolution model of the Indiantown Run in Pennsylvania using the DHSVM. The tests included different compression levels, all in contrast with an uncompressed control configuration in which all state variables needed to be assumed independent from each other. The results show that the produced forecasts display statistically significant improvements according to probabilistic metrics while maintaining the same performance according to a deterministic one. The improvements were larger for intermediate levels of compression, as too much compression led to a loss in physical realism of the state representation, and too little probably resulted in an under-exploitation of variable co-dependencies. The increase in prediction accuracy came at an additional computational cost, so final users need to find their preferred balance between accuracy and efficiency.

Multiple future work paths are possible using this new proximity coupling technique. For one, the clustering steps could be improved through the incorporation of additional variables, such as topographical features, into the distance function; or made more efficient through massive parallelization using GPUs. Moreover, distance functions that are not necessarily linear could be constructed to account for more complex interactions between variables, and additional clustering approaches could be tested. The partitioning of the watershed into super-pixels could also be replaced with an equivalent one-dimensional hillslope approach. Additionally, the method could be combined, for further efficiency, with distance decoupling methods—so that weak interactions are omitted from compressed states. Finally, the method could also be of utility in other geosciences, in which the level of spatial detail in the models is higher than in hydrology.

7.0 High-Resolution Model Initialization II: Efficient Variable Dependencies

This chapter presents a second alternative for reducing the dimensionality of state variable representations to help in the process of data assimilation. The chapter is based on the article “Lightweight Gaussian graphical modeling for efficient stochastic high-dimensional estimation: Application to hydrologic forecasting” by Felipe Hernández and Xu Liang, to be submitted to the Water Resources Research journal, currently in preparation.

7.1 Introduction

Whereas because of our limited understanding of the world, our limited observation capacity, or because of the inherently chaotic nature of geophysical phenomena, uncertainty will always be present when attempting to estimate unobserved conditions, predict the future, or run what-if decision-making scenarios. Geoscientists and engineers have thus embarked on a decades-long pursuit of probabilistic modeling tools that can adequately capture the extent of these gaps in our knowledge, thanks to the mounting evidence that better decisions can be made when there is an adequate measure of uncertainty. For example, meteorological agencies worldwide have long shifted to ensemble approaches for weather forecasting [232], and hydrologists have seen the benefits of stochastic flood simulations [172]. It is often the case that a probabilistic approach can be more valuable than a more-detailed deterministic one [171].

However, the Holy Grail of modeling, a probabilistic method that is both highly expressive and highly efficient, remains elusive. There exist evident tradeoffs between the detail level (given by their spatiotemporal resolution) or mathematical expressiveness in models and the ease of using them in a probabilistic framework. The standard practice of the popular Monte Carlo philosophy, for instance, leads to the need of running models numerous times, with different initial conditions on each occasion, so that the vast range of possible outcomes is sufficiently explored [233]. Moreover, many strategies require using these ensembles to determine a joint multivariate probability distribution that is also intended to capture the range of variation of the interactions between variables. Multivariate Gaussian probability distributions offer one of the simplest options to perform this probabilistic interpretation, as they only encode linear relationships between the covariates. Even with this restriction, storing the $p \times p$ covariance matrix $\mathbf{C}$ has quadratic complexity $\mathcal{O}(p^2)$ and extracting information from it (doing "inference") has cubic complexity $\mathcal{O}(p^3)$, and this extraction requires that there are more samples n than variables p—which is often not the case.

The field of data assimilation [234] is at the heart of modern geophysical prediction [235], [236]. Several of the techniques used to assimilate data use such Monte Carlo/Gaussian hybrids, and these constraints have inspired multiple alternatives to try to make these methods applicable. Some researchers have worked on reducing the dimensionality n of models in a way that minimizes loss of expressiveness [227], [229]. Others have produced lower-rank versions of $\mathbf{C}$ [224] or related matrices that are easier to operate on [225], [237]. On the other hand, many of the covariance values in $\mathbf{C}$ do not represent direct interactions between variables but are rather products of indirect relationships with others. Following this precept, it has also been suggested that considering all values is not necessary, and that focusing on those that really are can lead to faster computations [176].

This simplification is the core principle in the development of Gaussian graphical models (GGM) [238]. This family of machine learning methods aims at creating sparse versions (with many entries substituted by zeros) of either $\mathbf{C}$ or its inverse $\mathbf{C^{-1}}$ from a set of n samples, and efficient algorithms to ease their learning and—ideally—inference operations. While part of the promise of GGMs is to allow for improved tractability of Gaussian distributions, the great majority of the existing research has focused instead on the other goal of discovering cleaner representations of the underlying structure of the data—a path that has nonetheless proven useful in various applications.

A salient example is the graphical lasso [212], which is based on the application of one-vs-all regularized linear regressions for each variable [239]. The method uses the empirical covariance matrix $\boldsymbol{S}$ as a starting point and further requires multiple matrix inversions, which become overly expensive for high-dimensional problems. Despite having been found to be often equivalent to a simple thresholding technique [240], the popularity and appeal of the graphical lasso inspired multiple developments that brought improvements to both the accuracy and efficiency of the learning process through pseudo-likelihood methods [241]–[245] and alternative optimization approaches [246]–[249]. Techniques with other guiding principles, like Bayesian methods [250]–[252] or shrinkage operators [253], have also been proposed.

An interesting research path in GGMs is the partitioning of the variables into groups (or blocks) of related variables as a divide-and-conquer strategy to reduce the number of matrix operations [254]–[258]. This technique often utilizes two phases: one which determines the blocks, and another which imbues sparsity in the covariance matrix of each block [259], [260]. These approaches have the advantage that inference operations, such as computing conditional densities or generating random samples, require inverting much smaller matrices. GGMs in general can also potentially benefit from their imposed sparsity by accelerating these operations through the use of belief propagation techniques [261], but only scarce work has been performed on their application [262], [263].

This chapter introduces GGMLite (for lightweight GGM) as an answer to the need for efficient probabilistic models that are not limited to the discovery of structures in data. Like with multivariate Gaussian distributions, GGMLite includes tools for generating random samples, and determining the density and likelihood of arbitrary points—a feature that is essential in data assimilation and other geophysical applications. It also allows for efficient learning and inference by grouping variables together into overlapping blocks or "cliques" [264] with size dictated by the number of samples n available. Our technique includes the following additional differentiating points with existing methods: 1) a heuristic learning approach that disposes of the need for iterative optimization; 2) the absence of any matrix larger than $n \times n$; 3) the gradual construction of covariance matrices, a practice that curbs inversion costs; 4) the comfort of not having to calculate all the covariance terms in $\boldsymbol{S}$; 5) the omission of perfectly independent or perfectly linearly-dependent variables from the cliques, in order to maximize matrix compactness; and 6) the grouping of variables based on multivariate dependency metrics and not on one-to-one relationships.

The Methods section describes the structure of GGMLite and its learning, sampling, and density/likelihood algorithms; and its coupling with a data assimilation algorithm to allow for high-resolution geophysical forecasts. The Experimental setup section describes a high-resolution hydrological application with a physically-based modeling engine selected to test the enhanced assimilator. The Results and discussion section shows the comparison between the modified and unmodified forecasts and explores the effects of different parameterizations of the method. Finally, conclusions and future work guidelines are provided in the last section.

7.2 Methods

Lightweight Gaussian Graphical Models (GGMLite) are used to describe the joint probability distribution of a vector of variables $\boldsymbol{x} \in \mathbb{R}^p$. An instance of GGMLite is comprised of the following data elements: the mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\sigma}$ of each variable x, a list of variables that are constant ($\sigma_j = 0$), a list of variables that are unconditionally independent or "free," a list of regression coefficients for the variables that are a (perfect) linear combination of others ("determined"), and a set of cliques which contain overlapping groups of variables that are "covariates" with each other. Each clique k contains a covariance matrix of its m_k variables $\mathbf{C}_k$ and its corresponding Cholesky lower triangular matrix $\mathbf{L}_k$ (where $\mathbf{C}_k = \mathbf{L}_k\mathbf{L}_k^{\mathrm{T}}$). Variables cannot belong to several of these groups: they are either constant, free, determined, or covariates. While a single covariance matrix $\mathbf{C}$ can store all of this information, GGMLite opts for this type of compartmentalization of the information to minimize storage footprint (there might be cases in which the number of variables p is so large that $\mathbf{C}$ cannot even be instantiated).

7.2.1 Learning Algorithm

The following algorithm aims at creating an instance of GGMLite based on n samples $\boldsymbol{x}_i$ with dimensionality p. Ideally n would be large, but it is often the case that resource limitations lead to cases where $n \ll p$. n limits the rank of the corresponding covariance matrix $\boldsymbol{S}$ and, if there are not as many non-redundant samples (i.e., as many not co-hyperplanar) as p, $\boldsymbol{S}$ is not positive definite and cannot be inverted. In practical terms, this means that a few samples only contain enough information to fit a small linear model between their variables, whereas $\boldsymbol{S}$ is supposed to represent such a model with p degrees of freedom. While some GGM learning methods allow to sidestep this limitation—e.g., [241]—we believe that one should not attempt to extract more information than there is actually available at the risk of overfitting [76]. Therefore, clique size in GGMLite is limited by the ability to invert their covariance matrix.

Table 18 List of Parameters in GGMLite's Learning Algorithm

Symbol	Description	Range	Used
$m_{\%}$	Target rate of m_k/n for all cliques	$\mathbb{R} \in (0,1)$	0.8
$m_{\max}$	Hard constraint for m_k; supersedes $m_{\%}$	$\mathbb{N} \geq 2$	Infinite
m_{share}	Min. percentage of variables to be shared between cliques	$\mathbb{R} \in [0,1)$	25%
r^2_{free}	Max. regression r^2 for free variables	$\mathbb{R} \in [0,1]$	0.25
σ^2_{free}	Max. explained percentage of σ^2 for free variables	$\mathbb{R} \in [0,1]$	0.1
r^2_{determ}	Min. regression r^2 for determined variables	$\mathbb{R} \in [0,1]$	0.99
σ^2_{determ}	Min. explained percentage of σ^2 for determined variables	$\mathbb{R} \in [0,1]$	0.99
rand	true: randomize order of variables; false: use natural order of variables		true
split	Clique splitting mode: random, fast, or accurate		fast

Table 18 shows the parameters of GGMLite's learning algorithm, including their description and range. $m_{\%}$ and $m_{\max}$ limit the size m of the cliques, while m_{share} determines the degree of overlap between them by defining the percent of variables in each that must also appear in other cliques. This overlap is meant to maintain dependencies between variables that were not sorted into the same clique [264]. r^2_{free} and r^2_{determ} are limits to the coefficient of determination r^2 between a variable's observations and a linear regression estimator based on others. σ^2_{free} and σ^2_{determ} and are limits to the percentage of a variable's variance that is explained by such an estimator. rand controls whether if the order of the variables in the samples is to be randomized before constructing the cliques and should be set to true if there is no preferred order.

The following steps summarize the process of learning an instance of GGMLite based on a set of samples (programming variables of the same type and the same color indicate they correspond to the same instance):

1. Compute σ_j for each variable and identify *constants* ($\sigma_j = 0$)
2. Populate *list of non-constant variables*; if rand, randomize order of *list*
3. Initialize *clique list* with a clique with the first variable in *list*, and another with the second
4. Extract next *variable* from *list*, from *queue* if *list* is empty, or go to 13 if *queue* is empty
5. Order *clique list* in ascending clique size order
6. Select first *clique* in *list*
7. Offer *variable* to *clique*
    a. If *variable* is found to be determined, store *regression coefficients* and go to 4
    b. Otherwise, compute *clique*'s resulting change in independence index
8. If *clique* is the last in *list*, go to 9; otherwise, select next *clique* in *list* and go to 7
9. If *variable* is free, add to *queue* (or to the *list of free variables* if *list* is empty) and go to 4
10. Select clique with largest change in independence index and commit *variable* to it
11. If *variable* needs to be shared, commit *it* to clique with second largest change in the index
12. Split any of the cliques that contain *variable* if they reached the target size; go to 4
13. Delete any cliques with only one variable and add their variables to the *list of free variables*
14. Return *constants*, *clique list*, *regression coefficients*, and *list of free variables*

In essence, cliques are constructed gradually by offering them variables one at a time (step 7) and determining in which one(s) the new variable would strengthen variable interdependencies the most (steps 10 and 11). A queue is used to allow for variables that are found to be unrelated at first to be recycled after cliques have been populated. Given these set of policies, variables at the beginning of the list are more likely to be found within cliques, and variables at the end are more likely to be found determined. Following this logic, important variables should be offered first so that the model reflects the way in which they affect others. Therefore, if users want to imbue this physical knowledge in the process, rand should be set to false and the variables ordered from most important to less important (step 2). However, while we did not test this extensively, we do not expect major changes in the accuracy and the efficiency of the model if built with a random variable order, as the important relationships should be represented regardless of which variables are thought of as "independent" and which as "dependent" from a mathematical point of view.

The level of variable interdependency within cliques is assessed through an independence index $\gamma = |\mathbf{C}_k|/\prod_j c_{k,j,j}$, defined as the ratio of the determinant of matrix $\mathbf{C}_k$ and the product of its diagonal elements. Through the use of γ, GGMLite allows grouping variables together based on the overall tightness of the group rather than only on pairwise correlations as in previous approaches.

The most involved step in the learning algorithm is the offering of variables to cliques (step 7). When a variable is offered, an extended copy of $\mathbf{C}_k$ ($\mathbf{C}_k^+$) is created that includes the covariance values $\boldsymbol{c}$ corresponding to the new variable. A Cholesky decomposition [211] then similarly creates an extended copy of $\mathbf{L}_k$ ($\mathbf{L}_k^+$). Note that this operation can be done by copying $\mathbf{L}_k$ and only computing the new terms at the bottom row—thus reducing the baseline complexity of $\mathcal{O}(m^2)$. $|\mathbf{C}_k^+| = |\mathbf{L}_k^+|^2 = \left(\prod_j l_{k,j,j}^+\right)^2$ is then computed with the diagonal elements of $\mathbf{L}_k^+$ ($l_{k,j,j}^+$) to determine

γ^+ and the corresponding change $\Delta\gamma = \gamma^+ - \gamma$. $\mathbf{L}_k$ can help find the regression coefficients $\boldsymbol{w}$—to estimate the new variable from the rest—from the function $\mathbf{C}_k\boldsymbol{w} = \boldsymbol{c}$ [212] with $\mathcal{O}(m^2)$ complexity, by first forward-solving $\boldsymbol{y}$ from $\mathbf{L}_k\boldsymbol{y} = \mathbf{c}$ and then backward-solving $\boldsymbol{w}$ from $\mathbf{L}_k^{\mathrm{T}}\boldsymbol{w} = \boldsymbol{y}$ (a lower-upper "LU" evaluation). This step has the largest complexity of the algorithm, making the algorithm's overall complexity depend on the number of times this step is invoked and the size of the clique at each time. The constant coefficient of the regression is set as the difference between the mean of the offered variable and the dot product of the mean vector of the variables in the clique and the regression coefficients ($\boldsymbol{\mu}_k \cdot \boldsymbol{w}$). Finally, the regression can be applied to all samples to compute the percentage of explained variance and the coefficient of determination r^2.

For a variable to be classified as determined (step 7a), the linear estimator must have an r^2 of at least r^2_{determ} and must explain at least σ^2_{determ} of the variable's variance. Once a variable is found to be determined, any unexplained variance is also stored together with $\boldsymbol{w}$ and the constant coefficient. Conversely, for a variable to be classified as free (step 9), none of the linear estimators of the cliques should have an r^2 larger than r^2_{free} and explain more than σ^2_{free} of the variance. Variables that are neither constant, determined, or free are assigned to cliques as covariates (steps 10 and 11) by replacing $\mathbf{C}_k$ with $\mathbf{C}_k^+$ and $\mathbf{L}_k$ with $\mathbf{L}_k^+$. Only two cliques are created initially (step 3) but, as they grow, they might eventually split (step 12) if they reach the maximum size of $\min(m_{\%}n, m_{\max})$. The splitting can be performed in three ways defined by the split parameter: randomly assigning variables to the two new cliques ("random"), splitting by respecting their order ("fast"), or by offering one variable at a time and committing it to where $\Delta\gamma$ is larger ("accurate"). $\Delta\gamma$ usually shrinks fast when the first variables are added and then slower, increasing the likelihood that smaller cliques are selected in step 10. This results in a tendency for the cliques of being of uniform size.

7.2.2 Random Sampling

Random sampling, or sometimes "simulation," consists of stochastically creating new samples that conform to the probability distribution being modeled. To create a random sample using GGMLite, variables are sampled in order depending on their category: first the covariates, then the determined, and finally those that are constant or free. Sampling the covariates from each clique is the same as sampling from a Gaussian distribution: the vector of random variables $\boldsymbol{s}_{\text{random}} = \boldsymbol{\mu} + \mathbf{L}\boldsymbol{v}$, where $\boldsymbol{v}$ is a vector of independent standard normal random values and $\mathbf{L}$ is a lower triangular matrix such that $\mathbf{L}\mathbf{L}^{\mathrm{T}} = \mathbf{C}$ (from a Cholesky decomposition). However, given that variables are shared between cliques, these cannot be sampled independently from each other. Once one clique is sampled, the variables that have been determined affect the distribution in the other cliques. Therefore, a Bayesian belief propagation technique must be used to first condition the distributions on the new information before it can be sampled [265].

The following steps summarize the algorithm for generating samples from a clique when a subset of its variables (here referred to as $\boldsymbol{x}_2$) is already known:

1. If there are no known values, set $\boldsymbol{\mu}_{\text{cond}} = \boldsymbol{\mu}$, set $\mathbf{C}_{\text{cond}} = \mathbf{C}$, and go to 9
2. Assign known values to *random sample*
3. Organize and assign mean values in $\boldsymbol{\mu}$ into $\boldsymbol{\mu}_1$ (unknowns) and $\boldsymbol{\mu}_2$ (knowns)
4. Organize and assign covariance values in $\mathbf{C}$ into four sub-matrices $\begin{bmatrix} \mathbf{C}_{11} & \mathbf{C}_{12} \\ \mathbf{C}_{21} & \mathbf{C}_{22} \end{bmatrix}$
5. Compute $\mathbf{C}_{22}^{-1}$ (the inverse of $\mathbf{C}_{22}$)
6. Compute deviations of known values from the mean: $\boldsymbol{d}_2 = \boldsymbol{x}_2 - \boldsymbol{\mu}_2$
7. Compute mean of conditional distribution: $\boldsymbol{\mu}_{\text{cond}} = \boldsymbol{\mu}_1 + \mathbf{C}_{12}\mathbf{C}_{22}^{-1}\boldsymbol{d}_2$
8. Compute covariance of conditional distribution: $\mathbf{C}_{\text{cond}} = \mathbf{C}_{11} - \mathbf{C}_{12}\mathbf{C}_{22}^{-1}\mathbf{C}_{21}$
9. Compute $\mathbf{L}_{\text{cond}}$ from a Cholesky transform of $\mathbf{C}_{\text{cond}}$

10. Generate vector of random standard normal values $\boldsymbol{v}$

11. Generate random values: $\boldsymbol{s}_{\text{cond}} = \boldsymbol{\mu}_{\text{cond}} + \mathbf{L}_{\text{cond}}\boldsymbol{v}$

12. Add values in $\boldsymbol{s}_{\text{cond}}$ to *random sample* and return *random sample*

The algorithm has a temporal complexity dictated by step 5, in which the inverse of a covariance matrix of the known variables needs to be calculated. If a Cholesky decomposition is used as mentioned before, this can be accomplished in cubic time. Once all cliques have been sampled, samples for the determined variables can be computed by simply adding the constant term to the dot product between the vector of regression coefficients for that variable and the vector of sampled values for the corresponding covariates. Finally, free variables are sampled independently using univariate Gaussian sampling. Once all values have been determined, the full sample vector is assembled by organizing them according to the variable order.

7.2.3 Density and Likelihood Computation

Each clique in GGMLite effectively represents a multivariate Gaussian probability distribution. The probability density p at an arbitrary point $\boldsymbol{x}$, that matches the variables in the clique, can be computed as:

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^{n} \cdot |\mathbf{C}|}} \exp\left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^{\mathrm{T}} \mathbf{C}^{-1} (\boldsymbol{x} - \boldsymbol{\mu})\right] \tag{27}$$

For free variables, represented as univariate Gaussian distributions with mean μ and standard deviation σ, the probability density can be computed as:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right] \tag{28}$$

The density for a constant or a dependent variable can simply be defined as one if the value is the same as the variable, or zero if it is not. The probability density of an arbitrary point $\boldsymbol{x}$ given a full GGMLite instance can thus be computed as the product of the densities for all cliques and all free variables. However, for variables that were classified as constant, the values need to match exactly the means observed during the learning process. Similarly, for variables that were classified as dependent, the values need to match exactly those that are produced when solving the regression equation. If at least one of them does not match, the density is zero.

The density has problematic properties in order to be used effectively in many applications. For instance, it only takes one value for a constant variable that is misaligned for it to become zero, independently of how well the rest of the sample conforms to the full distribution. Moreover, given its multiplicative formulation, even with modest numbers of variables the density can quickly become extremely small (and eventually converge to zero) or extremely large (and eventually converge to infinity). For these reasons, we found it much better to use a likelihood function with better properties. We therefore propose using a likelihood metric based on the Mahalanobis distance, which is a multivariate generalization of one-dimensional distances that also takes into account inter-variate covariance values. For a clique, the squared Mahalanobis distance of $\boldsymbol{x}$ is given by:

$$d_{\text{Mahalanobis}}(\boldsymbol{x}) = \sqrt{(\boldsymbol{x} - \boldsymbol{\mu})^{\mathrm{T}}\mathbf{C}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})} \qquad (29)$$

For a free variable it is given by:

$$d_{\text{Mahalanobis}}(x) = \frac{x-\mu}{\sigma} \qquad (30)$$

Eq. (30) is also used to compute the distances to determined variables, using σ equals to a tiny constant ϵ when the sample standard deviation is zero. The likelihood of a full arbitrary point $\boldsymbol{x}$ given a GGMLite model is thus the inverse of the sum of the square distances for all cliques, all free variables, and all determined variables. When kernel density is used, the likelihood is computed by averaging the individual likelihood for each kernel computed in this way. However, when $\boldsymbol{x}$ is located exactly where one of the root samples is, the likelihood yields infinite because the distance is zero. To avoid this situation, which is not uncommon, and to allow to differentiate between samples located at the roots, we implemented an alternate averaging that ignores the kernel with the smallest distance. This alternative is called “leave-one-out” (loo) likelihood.

7.3 Results and Discussion

We implemented a GGMLite engine and coupled it with OPTIMISTS to allow for high-resolution data assimilation. The modified assimilator modifies the sampling step and the evaluation step as follows. The state probability distribution $\boldsymbol{S}$ is now modeled using GGMLite instead of multivariate Gaussian distributions and, if kernels are used, the kernels are now similarly represented by GGMLite instances instead of Gaussian ones. In the sampling step, random samples are created using the sampling algorithm described before. Similarly, if a secondary objective for judging candidate particles that measures deviations from the current distribution is used, the likelihood computation described above is used instead of the original one.

We designed a factorial experiment to test whether if GGMLite represents a better alternative than using diagonal covariance Gaussian distributions (in which all variables are assumed to be independent from each other). Additionally, the experiment was designed to determine if using kernels was advantageous over using unimodal distributions and, following from results in previous chapters, to test a variety of assimilation time steps Δt and two possible settings that affect the allowed diversity of the particle ensemble.

As before, the experiment was executed on the Indiantown Run model on the DHSVM by performing continuous streamflow forecasts at different lead times throughout a three-month window. A total of 144 test cases were run: 3 covariance options (diagonal, GGMLite, GGMLite with leave-one-out likelihood) × 2 modality options (mono, kernels) × 2 diversity options (not greedy, greedy) × 3 time steps Δt (6 hours, 2 days, and 1 week) × 4 lead times (6 hours, 1 day, 4 days, 16 days). The GGMLite parameters used are listed in Table 18. The values for these parameters were selected using trial and error and informed best guesses, and are certainly prone to being optimized. Additional OPTIMISTS parameter assignments are $n = 50$ samples, $p_{\text{samp}} = 1.0$, 2 objectives (MAE, likelihood), and kernel scaling = 0.1 (the factor to multiply the sample covariance matrices $\mathbf{C}$ to obtain the kernels bandwidth matrices $\mathbf{B}$).

Table 19 shows the p-values of the ANOVA of the experiment results for five tested forecast performance metrics. In all cases the factors have statistically significant effects on the performance, except for the modality on the density. There are also many factor interactions that have significant effects. Most performance metrics have large absolute correlations between them, indicating that, for example, if a configuration has a high NSE_{l2} it also has a low CRPS. However, the density did not show a high correlation with any of the other metrics, suggesting a certain degree of orthogonality between a forecast deterministic quality and its probabilistic quality. For example, the mean of the forecast can have relatively small deviations from the actual streamflow, but its spread might be too high (overconfidence) or too low (underconfidence).

Table 19 p-values From the ANOVA for Different Forecast Performance Metrics

Source	NSE$_{l2}$	NSE$_{l1}$	MARE	CRPS	Density
Model	0.000	0.000	0.000	0.000	0.000
Lead time (blocks)	0.000	0.000	0.000	0.000	0.003
Linear	0.000	0.000	0.000	0.000	0.000
Greed	0.000	0.000	0.000	0.000	0.000
Modality	0.000	0.000	0.000	0.000	0.842
Covariance	0.000	0.000	0.000	0.000	0.000
Δt	0.000	0.000	0.000	0.000	0.001
2-Way Interactions	0.000	0.000	0.000	0.000	0.000
Greed×Modality	0.041	0.035	0.033	0.009	0.328
Greed×Covariance	0.000	0.000	0.000	0.000	0.224
Greed× Δt	0.000	0.000	0.000	0.000	0.017
Modality×Covariance	0.000	0.000	0.001	0.000	0.014
Modality× Δt	0.019	0.011	0.001	0.104	0.680
Covariance× Δt	0.000	0.000	0.000	0.000	0.000
3-Way Interactions	0.000	0.000	0.000	0.000	0.000
Greed×Modality×Covariance	0.018	0.094	0.105	0.006	0.739
Greed×Modality× Δt	0.413	0.716	0.232	0.141	0.229
Greed×Covariance× Δt	0.000	0.000	0.000	0.000	0.098
Modality×Covariance× Δt	0.252	0.168	0.091	0.012	0.000
4-Way Interactions	0.000	0.016	0.222	0.004	0.248
Greed×Modality×Covariance× Δt	0.000	0.016	0.222	0.004	0.248

Figure 28 shows the main effect plots for the CRPS, which should roughly reflect the effects on the deterministic metrics as well, and for the density. The CRPS is reduced (better forecast) by using the greedy configuration, using kernels, using diagonal covariance matrices, and by selecting an intermediate assimilation time step. The density is increased (better forecast) also when the greedy configuration is used and with the intermediate time step, but not when using diagonal matrices instead of GGMLite. Only small differences are observed between the original and the leave-one-out versions of the likelihood estimation approach.

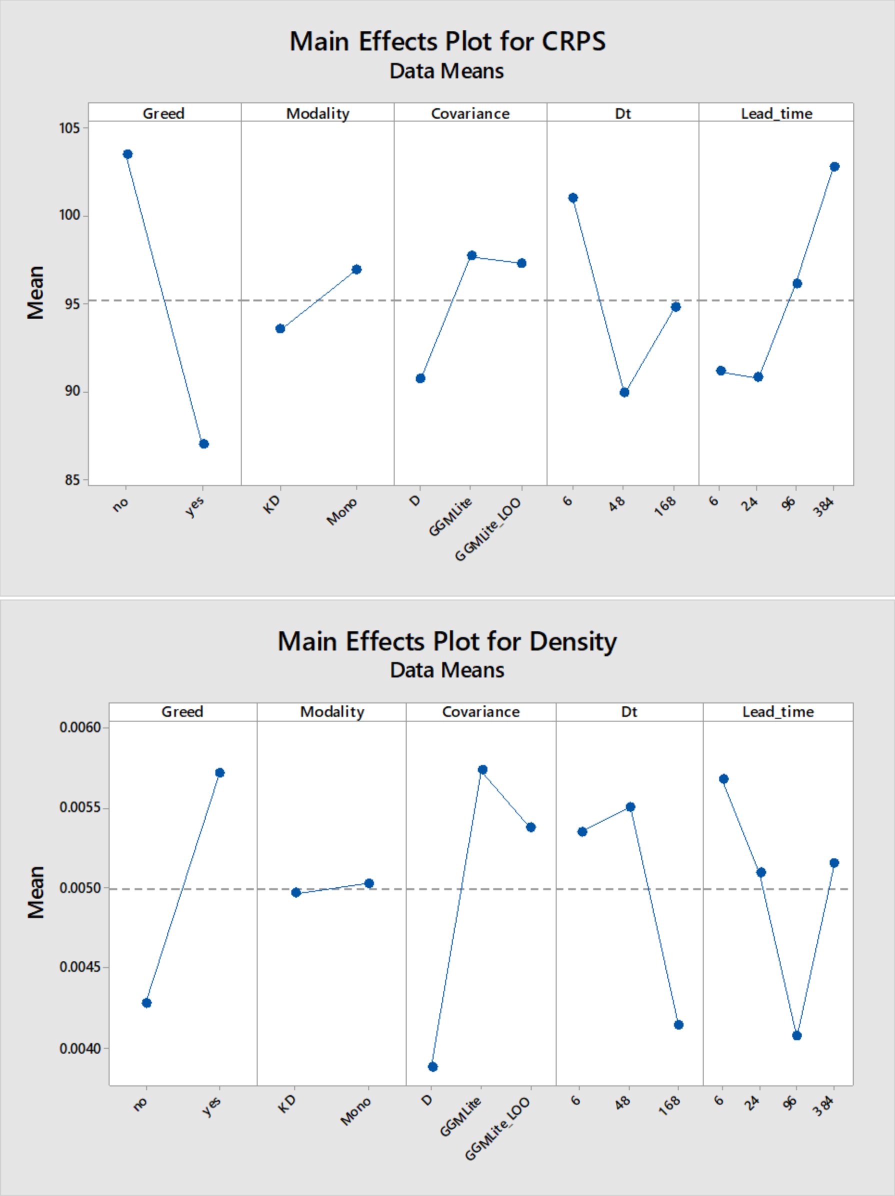


Figure 28 Main Effect Plots on Forecast CRPS and Density

Top: CRPS (l/s), bottom: Density.

These results suggest that there is a tension between the deterministic and the probabilistic qualities of a forecast, with GGMLite leading to better probabilistic performance according to the density. It has been shown that probabilistic forecasts that adequately capture the uncertainty of the estimates are usually preferable to deterministic forecasts of higher accuracy, given that they allow for fewer false alarms and fewer missed events in threat monitoring applications [171]. While this perspective gives further validity to the observed benefits of GGMLite in hydrologic data assimilation, we propose that both diagonal covariance matrices and GGMLite configurations be tested in future applications, and the tradeoffs between forecast performance metrics be analyzed for the specific conditions found therein.

7.4 Conclusions and Future Work

In this chapter we introduced Lightweight Gaussian graphical models, or GGMLite, a novel and efficient approach to representing multivariate Gaussian distributions and extracting information from them in cases of high-dimensionality and/or limited sample availability. GGMLite avoids instantiating and operating on full covariance matrix by grouping related variables together in a set of interdependent cliques. We provided algorithms for learning, sampling, and evaluating densities and likelihoods in GGMLite instances that make efficient use of the processing and storage resources. The learning algorithm in particular offers advantages over those in previous variable-grouping GGMs because the clustering approach measures the dependencies of all the variables in the cliques simultaneously instead of relying on metrics that only take into account pairs of variables. While the heuristic method does not guarantee an optimal clique arrangement, this feature procures the selection of strong relationships in a very efficient manner—compared with the iterative approach of standard optimization-based methods.

We tested a few configurations of GGMLite in the context of hydrologic forecasting by coupling the algorithm with the OPTIMISTS data assimilation method to produce streamflow estimates for a high-resolution model using limited observations and the DHSVM modeling engine. The results showed that GGMLite offers significant improvements to the probabilistic quality of the produced forecasts. Before further applications, it would be worthwhile to test GGMLite in a more generic setting by using it to approximate known probability distributions. Samples from the original distribution and from the fitted GGMLite instance could be analyzed using empirical cumulative functions [266], sample pairing [267], binning [268], or principal component analyses [269], [270]; to determine how well GGMLite represents the target distribution for different configurations, and how does the performance change.

Future work should continue to explore efficient alternatives to improve the algorithms introduced here, and to add further functionality to GGMLite—starting with the capability of estimating conditional distributions given discrete and stochastic observations of some of the variables. Moreover, additional applications for GGMLite in other fields should be explored, in the geosciences and beyond, given the approach's relative efficiency compared to other GGM methods and its available modeling scope that is not limited to learning. Within the field of hydrologic forecasting, future work should explore combining GGMLite with other methods for high-resolution data assimilation (as the clustering approach introduced in the previous chapter), and deployments in more comprehensive estimation frameworks that deal with uncertainty in model structures and parameters. Moreover, while limited by only being able to represent linear relationships, GGMLite's probabilistic nature might prove an interesting alternative or complement to physically-based models in probabilistic prediction. Since GGMLite does not require using Monte Carlo simulations, it would be able to provide stochastic estimates straightforwardly and efficiently in their stead.

8.0 Unified Hydrologic Estimation

This final chapter explores integration strategies for all the tools presented in this dissertation and provides an example of a complex hydrologic modeling workflow that could benefit from them.

8.1 Simultaneous Estimation of Parameters and Initial States

In order to further increase the accuracy of hydrologic predictions, researchers have long explored possible alternatives for addressing the uncertainty in both parameters and state variables simultaneously [177]. From one point of view, the difference in time frames and ranges in which parameters and states change favor a separate or sequential treatment of the two problems. That is, as we have done in this dissertation so far, calibration should be performed first using a long time period and data assimilation should be performed afterwards using a short period preceding the start of the forecast and assuming that parameters remain static. Gradual changes in the parameters can be accounted for by periodically re-calibrating the models using the most recent information.

On the other hand, if we allow the possibility that parameters can have rapid changes; maybe thanks to the interaction with variables such as temperature and soil moisture, and albeit not as large as those seen in state variables; a second position, where parameters and states are estimated simultaneously shortly before forecasts simulations are run, becomes attractive [271]–[273]. Even if one does not subscribe to the idea that model parameters are prone to relatively rapid changes, this type of approach can be framed from a more pragmatic perspective: it is not that the properties of the watershed necessarily fluctuate constantly, but that we are not sure of their actual values and thus we allow them to change as a way to compensate for both our lack of knowledge and for the more deeply-rooted errors that originate in the structures of our models. However, extending the vector of unknowns to incorporate model parameters in a data assimilation framework by necessity increases the tendency towards overfitting, as evidenced by the finding that parameters do not often reach a stable behavior [136]. This problem also becomes more acute for high-dimensional models.

Alternative perspectives on this matter exist, as is the case of some works that argue that state estimates could be incorporated within calibration frameworks rather than parameters within data assimilation frameworks [136], [274]. Such position can be defended by pointing to the fact that, during the calibration period, biases in the states can exist as much as they do when preparing forecasts, and that ultimately the parameter estimates illegitimately absorb these uncertainties unless the fallibility of state estimates is also embraced during the calibration stage. However, these approaches require running the expensive simulation schemes of data assimilation on the much longer time scale associated with calibration approaches, resulting in very high computational costs.

Following these ideas, we propose to integrate parameter and state variable estimation by combining the calibration methods presented in Chapters 2 and 3 with the OPTIMISTS data assimilation algorithm, presented in Chapters 4 through 7. The first step consists in obtaining a probabilistic prior estimate of the model's parameters to counter the underestimation of their uncertainty. For this effect, we propose adopting a probabilistic interpretation of the Pareto results obtained in the initial calibration process. As is done in OPTIMISTS, solutions in the final population are taken as samples of a nonparametric multivariate probability distribution, with front rankings used to weight them. Following the equifinality thesis [75], the natural spread occurring in the Pareto front due to tensions between the multiple objectives should provide a good approximation to the uncertainty of the estimates.

Once this prior parameter distribution has been established, by running the calibration algorithm for an extended period of time, OPTIMISTS obtains the joint distribution of parameters and initial states by assimilating observations in a reduced time window (prior to the beginning of the prediction period). A full set of four options, including three that make use of this parameter probability distribution are described here:

1. *Fully-static parameters*: A single set of parameter values is selected and maintained throughout all assimilation and prediction periods. This is the approach that has been used so far in the previous chapters.

2. *Semi-static parameters*: Even though parameters are allowed to change during the data assimilation period, the resulting posterior distribution (used for the prediction period) reverses to the prior distribution of parameters and the algorithm re-computes the state variables for all samples by conditioning the posterior distribution on the prior parameter distribution.

3. *Semi-dynamic parameters*: Event though the parameters are allowed to change during the data assimilation period, and the modified distribution is used during the prediction period, subsequent predictions reverse to the prior parameter distribution.

4. *Fully-dynamic parameters*: parameters are assumed to be able to vary as much as state variables, and thus are included in the state variable vector of the regular version of OPTIMISTS [271]–[273]. The modified parameters are used during the prediction period. Subsequent assimilation-prediction loops embed the changes made to the parameter distribution.

These four alternatives cover the range of how much state variables are to be allowed to change. Note that a mixed selection of the alternatives is possible and could be applied in cases where different groups of parameters have different natural rates of change. For example, parameters that change relatively fast can be managed using the static alternative, while parameters whose change occurs slowly can be managed using the fully-static alternative.

We performed a continuous forecasting test on the Indiantown Run, similar to the ones in previous chapters, to compare two alternatives: one with fully-static parameters and one with fully-dynamic parameters. In both cases kernel density GGMLite was used, with kernel scaling = 0.2, a greedy configuration, n = 50, and an assimilation time step of Δt = 2 days. Figure 29 shows the ≥50% maximum density band of the forecasted 24-hour lead-time hydrographs for both cases. The scores for the configuration with fully-static parameters are NSE_{l2} = 0.688, NSE_{l1} = 0.452, MARE = 26.14%, CRPS = 84.67 l/s, and density = 0.00462. The scores for the configuration with fully-dynamic parameters are NSE_{l2} = 0.801, NSE_{l1} = 0.602, MARE = 18.53%, CRPS = 48.77 l/s, and density = 0.00953. As can be seen, using dynamic parameters significantly increases the quality of the forecasts. In the graph this is evidenced by the reduction in bias and an increase in the spread of the forecast to better try to span the observations.

While the fully-static and fully-dynamic options have been explored in the literature before, we will investigate if the intermediate options listed above may provide increased accuracy, and whether if these improvements are related to the baseline ability of the model to provide adequate forecasts. For example, the Indiantown Run model has a relatively poor performance (Figure 9) which can presumably be offset by allowing for high variability in the parameter estimates. Would this approach be beneficial in a model that is better at generating forecasts? The biggest challenge in the implementation of these intermediate alternatives lies in the design of an inference algorithm for the GGMLite that is able to generate a conditional distribution of state variables given a joint parameter-state distribution and a sample of parameter values. As mentioned at the end of Chapter 7, this is a top priority in the list of future developments.

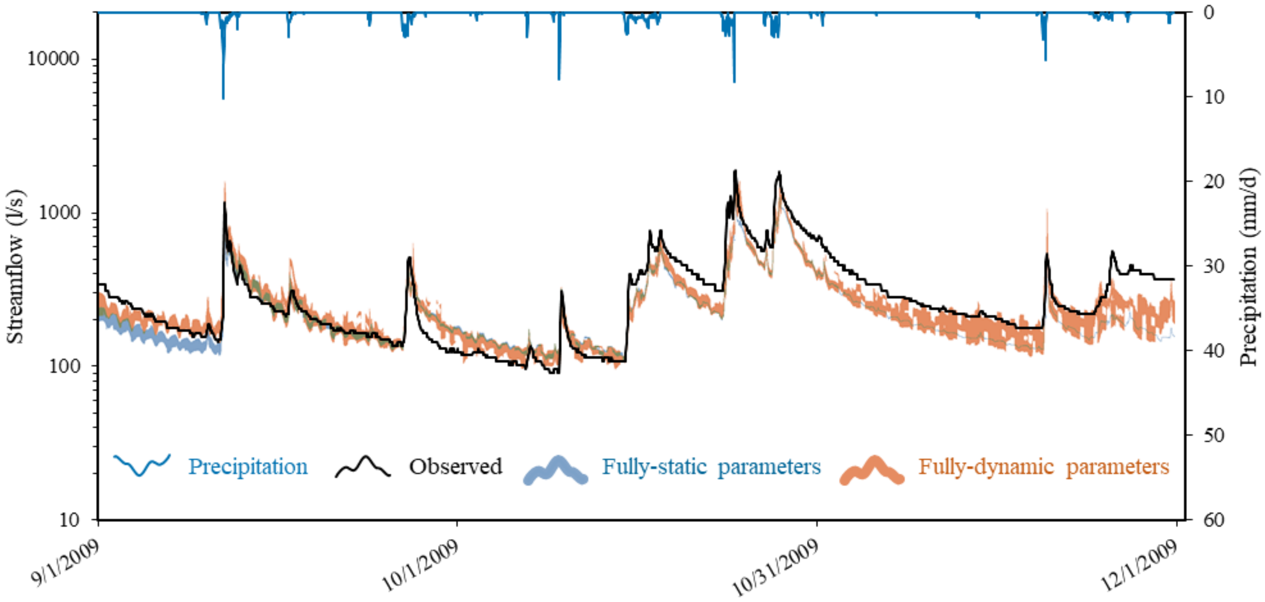


Figure 29 Probabilistic 24 Hour Forecast Comparisons for the Indiantown Run

The light blue and light orange bands illustrate the spread of the forecast by highlighting the areas where the probability density of the estimate is at least 50% of the density at the mode (the maximum) at that time step. The green bands indicate areas where the light blue and light orange bands intersect.

8.2 Summary and Perspectives

In the previous chapters we introduced the HDFR system, a unified environment to streamline complex hydrological modeling workflows comprising data acquisition, model creation and execution, and result analysis and export. Aligned with the objective of developing a single-stop platform for distributed data-based hydrologic modeling, we also introduced novel algorithms to optimize models for improved hydrologic prediction tasks. The dissertation addressed the challenge of model preparation by dividing the entire uncertainty into three distinct components: structural, parameter, and state variable uncertainty. For the latter two, new algorithms with multiple possible alternative configurations were proposed. Moreover, strategies for the coupling of these tools were also discussed, with the objective of outlining an integrated process for the holistic reduction of uncertainty in hydrologic modeling. Additionally, in order to allow tackling modern challenges related to large-scale hydrologic forecasting, especial emphasis was placed on the efficiency and scalability of the introduced techniques.

The following is a list of the original scientific contributions of this dissertation:

- Creation of a prototype GIS-based unifying system for hydro-meteorological data retrieval and optimized hydrological modeling.
- Extension of the popular gradient descent optimization paradigm to multi-objective global/evolutionary optimization problems.
- Incorporation of expert criteria into automatic hydrologic calibration processes for increased efficiency and parameter realism.
- Development of an efficient non-Gaussian, non-linear probabilistic data assimilation algorithm for high-resolution hydrologic forecasting applications.
- Simplification of probabilistic representations of the states of high-resolution models for accelerated hydrologic data assimilation.
- Implementation of a lightweight alternative to Gaussian graphical modeling with frugal computational requirements.
- Exploration of multiple alternatives to the simultaneous estimation of a model's parameters and initial conditions for optimized hydrologic forecasting.

As an application example of the tools presented in this dissertation, here we summarize a possible process for the continuous forecasting of the streamflow of a river of interest. The first part of the process is to be conducted offline and would consist of the following steps:

1. Acquisition of static watershed information (e.g., elevation, land cover, soil type map, etc.).
2. Acquisition of meteorological forcings for a predefined calibration period using the HDFR's data modules (e.g., precipitation, temperature, solar radiation, air humidity).
3. Acquisition of hydrological observations for the calibration period (e.g., discharge, top-layer soil moisture, snow cover).
4. Watershed and channel network delineation using HDFR's geomorphologic pre-processing.
5. Automatic creation of the model's files with the desired spatiotemporal resolution.
6. Probabilistic multi-objective calibration of the model's parameters using either the baseline optimization algorithm or HIP-POP.

The second part of the process is to be conducted online at predefined time intervals to provide a continuously-updating estimate of future conditions in the river:

1. Acquisition of meteorological forcings in the near past for the assimilation period.
2. Acquisition of hydrological observations in the near past for the assimilation period.
3. Dual initial state-parameter estimation using data assimilation through OPTIMISTS (with high-efficiency modifications depending on the dimensionality of the model).
4. Acquisition of forecasted meteorological forcings for the near future using the HDFR's forecast data modules.
5. Probabilistic ensemble forecast using the optimized model simulations.
6. Severity or additional application-specific analyses.
7. Result visualization and export.

As shown in this example, the tools described in this dissertation have the unprecedented capability of allowing for an integrated management of uncertainty in hydrologic modeling within a unified and lightweight framework. Moreover, the tools developed lend themselves easily towards the escalation to advanced computing infrastructure with massive parallelism, which would enable their application in large-scale operational settings. Finally, we see the potential of these tools being extended to additional applications in other Geosciences, such as Geology, Meteorology, and Oceanography.

Bibliography

[1] R. A. Fulton, J. P. Breidenbach, D.-J. Seo, D. A. Miller, and T. O'Bannon, "The WSR-88D Rainfall Algorithm," *Weather Forecast.*, vol. 13, no. 2, pp. 377–395, 1998.

[2] W. L. Barnes, T. S. Pagano, and V. V. Salomonson, "Prelaunch characteristics of the Moderate Resolution Imaging Spectroradiometer (MODIS) on EOS-AM1," *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 4, pp. 1088–1100, 1998.

[3] A. Y. Hou, G. Skofronick-Jackson, C. D. Kummerow, and J. M. Shepherd, "Global precipitation measurement," in *Precipitation: Advances in Measurement, Estimation and Prediction*, S. Michaelides, Ed. Springer Berlin Heidelberg, 2008, pp. 131–169.

[4] D. Entekhabi, E. G. Njoku, P. E. O'Neill, K. H. Kellogg, W. T. Crow, W. N. Edelstein, J. K. Entin, S. D. Goodman, T. J. Jackson, J. Johnson, J. Kimball, J. R. Piepmeier, R. D. Koster, N. Martin, K. C. McDonald, M. Moghaddam, S. Moran, R. Reichle, J. C. Shi, M. W. Spencer, S. W. Thurman, L. Tsang, and J. Van Zyl, "The soil moisture active passive (SMAP) mission," *Proc. IEEE*, vol. 98, no. 5, pp. 704–716, 2010.

[5] J. S. Whitaker, T. M. Hamill, X. Wei, Y. Song, and Z. Toth, "Ensemble Data Assimilation with the NCEP Global Forecast System," *Mon. Weather Rev.*, vol. 136, no. 2, pp. 463–482, 2008.

[6] E. Rogers, G. DiMego, T. Black, M. Ek, B. Ferrier, G. Gayno, Z. Janic, Y. Lin, M. Pyle, V. Wong, and W.-S. Wu, "The NCEP North American Mesoscale Modeling System: Recent Changes and Future Plans," *23rd Conf. Weather Anal. Forecast. Conf. Numer. Weather Predict.*, no. 1995, 2009.

[7] F. Molteni, R. Buizza, T. N. Palmer, and T. Petroliagis, "The ECMWF ensemble prediction system: Methodology and validation," *Q. J. R. Meteorol. Soc.*, vol. 122, no. 529, pp. 73–119, 1996.

[8] X. Liang, D. P. Lettenmaier, E. F. Wood, and S. J. Burges, "A simple hydrologically based model of land surface water and energy fluxes for general circulation models," *J. Geophys. Res.*, vol. 99, no. D7, p. 14415, 1994.

[9] M. B. Ek, K. E. Mitchell, Y. Lin, E. Rogers, P. Grunmann, V. Koren, G. Gayno, and J. D. Tarpley, "Implementation of Noah land surface model advances in the National Centers for Environmental Prediction operational mesoscale Eta model," *J. Geophys. Res.*, vol. 108, no. D22, pp. 1–16, 2003.

[10] M. S. Wigmosta, B. Nijssen, and P. Storck, "The distributed hydrology soil vegetation model," *Math. Model. Small Watershed Hydrol. Appl.*, pp. 7–42, 2002.

[11] F. Chen and J. Dudhia, "Coupling an Advanced Land Surface–Hydrology Model with the Penn State–NCAR MM5 Modeling System. Part I: Model Implementation and Sensitivity," *Mon. Weather Rev.*, vol. 129, pp. 569–585, 2001.

[12] S. Hong, V. Lakshmi, E. E. Small, F. Chen, M. Tewari, and K. W. Manning, "Effects of vegetation and soil moisture on the simulated land surface processes from the coupled WRF/Noah model," *J. Geophys. Res. Atmos.*, vol. 114, no. 18, pp. 1–13, 2009.

[13] D. Gochis, A. Parodi, R. Hooper, S. Jha, and I. Zaslavsky, "Advancing hydrometeorological prediction capabilities through standards-based cyberinfrastructure development: The community WRF-Hydro modeling system," in *EGU General Assembly Conference Abstracts*, 2013, vol. 15, p. 6011.

[14] J. A. Foley, I. C. Prentice, N. Ramankutty, S. Levis, D. Pollard, S. Sitch, and A. Haxeltine, "An integrated biosphere model of land surface processes, terrestrial carbon balance, and vegetation dynamics," *Global Biogeochemical Cycles*, vol. 10, no. 4. p. 603, 1996.

[15] P. L. Wilkinson, M. G. Anderson, and D. M. Lloyd, "An integrated hydrological model for rain-induced landslide prediction," *Earth Surf. Process. Landforms*, vol. 27, no. 12, pp. 1285–1297, 2002.

[16] E. R. Vivoni, G. Mascaro, S. Mniszewski, P. Fasel, E. P. Springer, V. Y. Ivanov, and R. L. Bras, "Real-world hydrologic assessment of a fully-distributed hydrological model in a parallel computing environment," *J. Hydrol.*, vol. 409, no. 1–2, pp. 483–496, 2011.

[17] C. H. David, D. R. Maidment, G.-Y. Niu, Z.-L. Yang, F. Habets, and V. Eijkhout, "River Network Routing on the NHDPlus Dataset," *J. Hydrometeorol.*, vol. 12, no. 5, pp. 913–934, 2011.

[18] R. M. Maxwell, "A terrain-following grid transform and preconditioner for parallel, large-scale, integrated hydrologic modeling," *Adv. Water Resour.*, vol. 53, pp. 109–117, 2013.

[19] D. Tristram, D. Hughes, and K. Bradshaw, "Accelerating a hydrological uncertainty ensemble model using graphics processing units (GPUs)," *Comput. Geosci.*, vol. 62, pp. 178–186, 2014.

[20] X. Liang, D. Salas, M. Navarro, Y. Liang, W. Teng, R. Hooper, P. Restrepo, and J. Bales, "A New Open Data Open Modeling Framework for the Geosciences Community," in *American Geophysical Union Fall meeting*, 2013, p. H11E–1204.

[21] M. P. Clark, M. F. P. Bierkens, L. Samaniego, R. A. Woods, R. Uijlenhoet, K. E. Bennett, V. R. N. Pauwels, X. Cai, A. W. Wood, and C. D. Peters-Lidard, "The evolution of process-based hydrologic models: Historical challenges and the collective quest for physical realism," *Hydrol. Earth Syst. Sci.*, vol. 21, no. 7, pp. 3427–3440, 2017.

[22] R. D. Koster, A. K. Betts, P. A. Dirmeyer, M. Bierkens, K. E. Bennett, S. J. Déry, J. P. Evans, R. Fu, F. Hernández, L. R. Leung, X. Liang, M. Masood, H. Savenije, G. Wang, and X. Yuan, “Hydroclimatic variability and predictability: a survey of recent research,” *Hydrol. Earth Syst. Sci.*, vol. 21, no. 7, pp. 3777–3798, Jul. 2017.

[23] X. Chu and A. Steinman, “Event and Continuous Hydrologic Modeling with HEC-HMS,” *J. Irrig. Drain. Eng.*, vol. 135, no. 1, pp. 119–124, 2009.

[24] L. A. Rossman, *Storm water management model user’s manual, version 5.0*. National Risk Management Research Laboratory, Office of Research and Development, US Environmental Protection Agency Cincinnati, OH, 2010.

[25] W. H. Merkel, “Muskingum-Cunge Flood Routing Procedure in NRCS Hydrologic Models,” *Second Fed. Interag. Hydrol. Model. Conf.*, vol. 20, no. July, pp. 1–12, 2002.

[26] T. L. Jantzen, W. Resources, P. Technologist, D. M. Mcclelland, W. Resources, S. Technologist, J. A. Eichler, S. Engineer, L. Colorado, R. Authority, N. M. Gullo, S. Engineer, D. Lead, and L. C. River, “Use of Flood Modeller Pro to Develop Linked , Alternating 1-D and 2-D Models of Overland and in-River Flows for Breach of a Large Off-Channel Ring Dam,” in *Proceedings of the 32nd annual conference of the Association of State Dam Safety Officials*, 2015, pp. 1–18.

[27] P. F. Boulos, M. K. Muleta, C.-H. Orr, and J. J. Ro, “An innovative geocentric decision support solution to comprehensive planning, design, operation, and management of urban drainage systems,” *World Environ. Water Resour. Congr. 2006 Examining Conflu. Environ. Water Concerns*, pp. 1–10, 2007.

[28] A. L. I. Erturk, M. Gurel, M. A. Baloch, T. Dikerler, E. Varol, N. Akbulut, and A. Tanik, “Application of watershed modeling system (WMS) for integrated management of a watershed in Turkey,” *J. Environ. Sci. Heal. Part A*, vol. 41, no. 9, pp. 2045–2056, 2006.

[29] G. B. Bonan, P. J. Lawrence, K. W. Oleson, S. Levis, M. Jung, M. Reichstein, D. M. Lawrence, and S. C. Swenson, “Improving canopy processes in the Community Land Model version 4 (CLM4) using global flux fields empirically inferred from FLUXNET data,” *J. Geophys. Res.*, vol. 116, no. G2, pp. 1–22, 2011.

[30] D. N. Graham and M. B. Butts, “Flexible Integrated Watershed Modeling with MIKE SHE,” in *Watershed Models*, V. P. S. & D. K. Frevert, Ed. CRC Press, 2005, pp. 245–272.

[31] G. Y. Niu, Z. L. Yang, K. E. Mitchell, F. Chen, M. B. Ek, M. Barlage, A. Kumar, K. Manning, D. Niyogi, E. Rosero, M. Tewari, and Y. Xia, “The community Noah land surface model with multiparameterization options (Noah-MP): 1. Model description and evaluation with local-scale measurements,” *J. Geophys. Res. Atmos.*, vol. 116, no. 12, pp. 1–19, 2011.

[32] K. Beven and J. Freer, “A dynamic TOPMODEL,” *Hydrol. Process.*, vol. 15, no. 10, pp. 1993–2011, 2001.

[33] D. P. Ames, J. S. Horsburgh, Y. Cao, J. Kadlec, T. Whiteaker, and D. Valentine, "HydroDesktop: Web services-based software for hydrologic data discovery, download, visualization, and analysis," *Environ. Model. Softw.*, vol. 37, pp. 146–156, 2012.

[34] R. P. Hooper, M. Seul, J. Pollak, and A. Couch, "Realizing the potential of the CUAHSI Water Data Center to advance Earth Science," in *AGU Fall Meeting Abstracts*, 2015.

[35] D. G. Tarboton, R. Idaszak, J. S. Horsburgh, J. Heard, D. Ames, J. L. Goodall, L. Band, V. Merwade, A. Couch, and J. Arrigo, "HydroShare: advancing collaboration through hydrologic data and model sharing," *Int. Congr. Environ. Model. Softw.*, no. JUNE, 2014.

[36] M. A. Rajib, V. Merwade, I. L. Kim, L. Zhao, C. Song, and S. Zhe, "SWATShare - A web platform for collaborative research and education through online sharing, simulation and visualization of SWAT models," *Environ. Model. Softw.*, vol. 75, pp. 498–512, 2016.

[37] G. R. Pearce, T. K. Millard, and A. Harper, "Application of OpenMI interfacing to promote integrated modelling across the water/environment sector-the FluidEarth platform," no. May, 2010.

[38] J. B. Gregersen, P. J. a. Gijsbers, and S. J. P. Westen, "OpenMI: Open modelling interface," *J. Hydroinformatics*, vol. 9, p. 175, 2007.

[39] S. D. Peckham, E. W. H. Hutton, and B. Norris, "A component-based approach to integrated modeling in the geosciences: The design of CSDMS," *Comput. Geosci.*, vol. 53, pp. 3–12, 2013.

[40] S. D. Peckham and J. L. Goodall, "Driving plug-and-play models with data from web services: A demonstration of interoperability between CSDMS and CUAHSI-HIS," *Comput. Geosci.*, vol. 53, pp. 154–161, 2013.

[41] O. David, J. C. Ascough II, G. H. Leavesley, and L. R. Ahuja, "Rethinking modeling framework design: Object Modeling System 3.0," *Model. Environ. Sake Proc. 5th Bienn. Conf. Int. Environ. Model. Softw. Soc. iEMSs 2010*, vol. 2, pp. 1190–1198, 2010.

[42] C. Hill, C. DeLuca, Balaji, M. Suarez, and A. Da Silva, "The architecture of the Earth system modeling framework," *Comput. Sci. Eng.*, vol. 6, no. 1, pp. 18–28, 2004.

[43] M. Werner, J. Schellekens, P. Gijsbers, M. van Dijk, O. van den Akker, and K. Heynert, "The Delft-FEWS flow forecasting system," *Environ. Model. Softw.*, vol. 40, pp. 65–77, 2013.

[44] M. S. Wigmosta, L. W. Vail, and D. P. Lettenmaier, "A distributed hydrology-vegetation model for complex terrain," *Water Resour. Res.*, vol. 30, no. 6, pp. 1665–1679, 1994.

[45] M. Neteler and H. Mitasova, *Open source GIS: a GRASS GIS approach*. Springer Science & Business Media, 2013.

[46] F. P. Donnelly, “Evaluating open source GIS for libraries,” *Libr. Hi Tech*, vol. 28, no. 1, pp. 131–151, 2010.

[47] D. Chen, S. Shams, C. Carmona-Moreno, and A. Leone, “Assessment of open source GIS software for water resources management in developing countries,” *J. Hydro-Environment Res.*, vol. 4, no. 3, pp. 253–264, 2010.

[48] S. Steiniger and E. Bocher, “An overview on current free and open source desktop GIS developments,” *Int. J. Geogr. Inf. Sci.*, vol. 23, no. 10, pp. 1345–1370, 2009.

[49] S. Steiniger and G. Hay, “Free and open source geographic information tools for landscape ecology,” *Ecol. Inform.*, no. July, pp. 1–26, 2009.

[50] P. Zambelli, S. Gebbert, and M. Ciolli, “Pygrass: An Object Oriented Python Application Programming Interface (API) for Geographic Resources Analysis Support System (GRASS) Geographic Information System (GIS),” *ISPRS Int. J. Geo-Information*, vol. 2, no. 1, pp. 201–219, 2013.

[51] Y. Lin and K. E. Mitchell, “The NCEP Stage II/IV hourly precipitation analyses: development and applications,” in *19th Conf. on Hydrology, American Meteorological Society*, 2005, pp. 2–5.

[52] B. A. Cosgrove, D. Lohmann, K. E. Mitchell, P. R. Houser, E. F. Wood, J. C. Schaake, A. Robock, C. Marshall, J. Sheffield, Q. Duan, L. Luo, R. W. Higgins, R. T. Pinker, J. D. Tarpley, and J. Meng, “Real-time and retrospective forcing in the North American Land Data Assimilation System (NLDAS) project,” *J. Geophys. Res. Atmos.*, vol. 108, no. D22, pp. 1–12, Nov. 2003.

[53] C. Kummerow, W. Barnes, T. Kozu, J. Shiue, and J. Simpson, “The Tropical Rainfall Measuring Mission (TRMM) sensor package,” *J. Atmos. Ocean. Technol.*, vol. 15, no. 3, pp. 809–817, 1998.

[54] J. L. Goodall, J. S. Horsburgh, T. L. Whiteaker, D. R. Maidment, and I. Zaslavsky, “A first approach to web services for the National Water Information System,” *Environ. Model. Softw.*, vol. 23, no. 4, pp. 404–411, 2008.

[55] A. P. Barrett, *National operational hydrologic remote sensing center snow data assimilation system (SNODAS) products at NSIDC*. National Snow and Ice Data Center, Cooperative Institute for Research in Environmental Sciences, 2003.

[56] D. K. Hall, G. A. Riggs, V. V. Salomonson, N. E. DiGirolamo, and K. J. Bayr, “MODIS snow-cover products,” *Remote Sens. Environ.*, vol. 83, no. 1–2, pp. 181–194, 2002.

[57] P. A. Miller, M. F. Barth, L. A. Benjamin, R. S. Artz, and W. R. Pendergrass, “The Meteorological Assimilation and Data Ingest System (MADIS): Providing value-added observations to the meteorological community,” in *21st Conference on Weather Analysis and Forecasting*, 2005.

[58] P. Cornillon, J. Gallagher, and T. Sgouros, “Opendap: Accessing Data in a Distributed, Heterogeneous Environment,” *Data Sci. J.*, vol. 2, no. November, pp. 164–174, 2003.

[59] R. Rew and G. Davis, “NetCDF: An Interface for Scientific Data Access,” *IEEE Comput. Graph. Appl.*, vol. 10, no. 4, pp. 76–82, 1990.

[60] E. Waltz and J. Llinas, *Multisensor data fusion*, vol. 685. Artech house Boston, 1990.

[61] L. M. Parada and X. Liang, “Optimal multiscale Kalman filter for assimilation of near-surface soil moisture into land surface models,” *J. Geophys. Res. D Atmos.*, vol. 109, pp. 1–21, 2004.

[62] S. Wang, X. Liang, and Z. Nan, “How much improvement can precipitation data fusion achieve with a Multiscale Kalman Smoother-based framework?,” *Water Resour. Res.*, vol. 47, no. 3, 2011.

[63] D. E. Salas and X. Liang, “An Introduction to Multi-scale Kalman Smoother-Based Framework and Its Application to Data Assimilation,” in *Land Surface Observation, Modeling and Data Assimilation*, World Scientific, 2013, pp. 275–334.

[64] C. Svensson and D. A. Jones, “Review of rainfall frequency estimation methods,” *J. Flood Risk Manag.*, vol. 3, no. 4, pp. 296–313, 2010.

[65] S. Perica, D. Martin, S. Pavlovic, I. Roy, M. S. Laurent, C. Trypaluk, D. Unruh, M. Yekta, and G. Bonnin, “NOAA Atlas 14 Volume 9 Version 2, Precipitation-Frequency Atlas of the United States, Southeastern States,” *NOAA, Natl. Weather Serv. Silver Spring, MD*, p. 18, 2013.

[66] G. I. Snyder, “The 3D Elevation Program — Summary of Program Direction,” no. September, p. 2, 2012.

[67] C. Homer, J. Fry, and C. Barnes, “The National Land Cover Database,” *US Geol. Surv. Fact Sheet*, vol. 3020, no. February, pp. 1–4, 2012.

[68] D. A. Miller and R. A. White, “A Conterminous United States Multilayer Soil Characteristics Dataset for Regional Climate and Hydrology Modeling,” *Earth Interact.*, vol. 2, no. 1, pp. 1–26, 1998.

[69] R. B. Clapp and G. M. Hornberger, “Empirical equations for some soil hydraulic properties,” *Water Resour. Res.*, vol. 14, no. 4, pp. 601--604, 1978.

[70] E. D. Gutmann and E. E. Small, “A comparison of land surface model soil hydraulic properties estimated by inverse modeling and pedotransfer functions,” *Water Resour. Res.*, vol. 43, no. 5, pp. 1–13, 2007.

[71] V. Koren, F. Moreda, and M. Smith, “Use of soil moisture observations to improve parameter consistency in watershed calibration,” *Phys. Chem. Earth*, vol. 33, no. 17–18, pp. 1068–1080, 2008.

[72] W. W. Immerzeel and P. Droogers, “Calibration of a distributed hydrological model based on satellite evapotranspiration,” *J. Hydrol.*, vol. 349, no. 3–4, pp. 411–424, 2008.

[73] J. Parajka and G. Blöschl, “The value of MODIS snow cover data in validating and calibrating conceptual hydrologic models,” *J. Hydrol.*, vol. 358, no. 3–4, pp. 240–258, 2008.

[74] A. Efstratiadis and D. Koutsoyiannis, “One decade of multi-objective calibration approaches in hydrological modelling: a review,” *Hydrol. Sci. J.*, vol. 55, no. February 2015, pp. 58–78, 2010.

[75] K. Beven, “A manifesto for the equifinality thesis,” *J. Hydrol.*, vol. 320, no. 1–2, pp. 18–36, 2006.

[76] D. M. Hawkins, “The Problem of Overfitting,” *J. Chem. Inf. Comput. Sci.*, vol. 44, no. 1, pp. 1–12, 2004.

[77] F. Peng, K. Tang, G. Chen, and X. Yao, “Population-Based Algorithm Portfolios for Numerical Optimization,” *Evol. Comput. IEEE Trans.*, vol. 14, no. 5, pp. 782–800, 2010.

[78] Y. Wang, J. Huang, W. S. Dong, J. C. Yan, C. H. Tian, M. Li, and W. T. Mo, “Two-stage based ensemble optimization framework for large-scale global optimization,” *Eur. J. Oper. Res.*, vol. 228, no. 2, pp. 308–320, 2013.

[79] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *Evol. Comput. IEEE Trans.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.

[80] S. Droste, T. Jansen, and I. Wegener, “Optimization with randomized search heuristics—the (A)NFL theorem, realistic scenarios, and difficult functions,” *Theor. Comput. Sci.*, vol. 287, no. 1, pp. 131–144, 2002.

[81] J. A. Vrugt and B. A. Robinson, “Improved evolutionary optimization from genetically adaptive multimethod search,” *Proc. Natl. Acad. Sci.*, vol. 104, no. 3, pp. 708–711, 2007.

[82] J. A. Vrugt, B. A. Robinson, and J. M. Hyman, “Self-Adaptive Multimethod Search for Global Optimization in Real-Parameter Spaces,” *Evol. Comput. IEEE Trans.*, vol. 13, no. 2, pp. 243–259, Apr. 2009.

[83] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” *Evol. Comput. IEEE Trans.*, vol. 6, no. 2, pp. 182–197, 2002.

[84] M. Dorigo and T. Stützle, *Ant Colony Optimization*, 1st ed. Cambridge, USA: The MIT Press, 2004.

[85] H. R. Maier, A. R. Simpson, A. C. Zecchin, W. K. Foong, K. Y. Phang, H. Y. Seah, and C. L. Tan, “Ant Colony Optimization for Design of Water Distribution Systems,” no. June, pp. 200–209, 2003.

[86] M. López-Ibáñez, “Ant colony optimization for optimal control of pumps in water distribution networks,” *J. water Resour. …*, no. August, pp. 337–346, 2008.

[87] K. Socha and M. Dorigo, “Ant colony optimization for continuous domains,” *Eur. J. Oper. Res.*, vol. 185, no. 3, pp. 1155–1173, 2008.

[88] B. B. W. Silverman, *Density estimation for statistics and data analysis*, vol. 26, no. 1951. CRC press, 2018.

[89] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*, 2nd ed. John Wiley & Sons, 2015.

[90] H. Haario, E. Saksman, and J. Tamminen, “An Adaptive Metropolis Algorithm,” *Bernoulli*, vol. 7, no. 2, pp. 223–242, 2001.

[91] R. J. Hyndman, D. M. Bashtannyk, and G. K. Grunwald, “Estimating and Visualizing Conditional Densities,” *J. Comput. Graph. Stat.*, vol. 5, no. 4, p. 315, 1996.

[92] C. Dai, W. Chen, and Y. Zhu, “Seeker Optimization Algorithm for Digital IIR Filter Design,” *Ind. Electron. IEEE Trans.*, vol. 57, no. 5, pp. 1710–1718, 2010.

[93] J. D. Hewlett, B. M. Wilamowski, and G. Dundar, “Optimization using a modified second-order approach with evolutionary enhancement,” *Ind. Electron. IEEE Trans.*, vol. 55, no. 9, pp. 3374–3380, 2008.

[94] P. R. Seaber, F. P. Kapinos, and G. L. Knapp, *Hydrologic unit maps*. US Government Printing Office Washington, DC, USA, 1987.

[95] E. Rodríguez, C. S. Morris, and J. E. Belz, “A global assessment of the SRTM performance,” *Photogramm. Eng. Remote Sens.*, vol. 72, no. 3, pp. 249–260, 2006.

[96] J. G. Acker and G. Leptoukh, “Online analysis enhances use of NASA Earth science data,” *Eos, Trans. Am. Geophys. Union*, vol. 88, no. 2, pp. 14–17, 2007.

[97] D. C. Montgomery, *Design and analysis of experiments*, Eigth edit. John Wiley & Sons, 2012.

[98] J. E. Nash and J. V Sutcliffe, “River flow forecasting through conceptual models part I—A discussion of principles,” *J. Hydrol.*, vol. 10, no. 3, pp. 282–290, 1970.

[99] G. G. Villalba, F. Plaza, X. Zhong, W. T. Davis, M. Navarro, Y. Li, A. T. Slater, Y. Liang, X. Liang, P. Hydrology, Y. Li, T. A. Slater, Y. Liang, X. Liang, G. G. Villalba, F. Plaza, X. Zhong, W. T. Davis, M. Navarro, Y. Li, A. T. Slater, Y. Liang, and X. Liang, “A Networked Sensor System for the Analysis of Plot-Scale Hydrology,” *Sensors*, vol. 17, no. 3, pp. 1–27, 2017.

[100] M. J. Best, G. Abramowitz, H. R. Johnson, A. J. Pitman, G. Balsamo, A. Boone, M. Cuntz, B. Decharme, P. a. Dirmeyer, J. Dong, M. Ek, Z. Guo, V. Haverd, B. J. . van den Hurk, G. S. Nearing, B. Pak, C. Peters-Lidard, J. A. J. Santanello, L. Stevens, and N. Vuichard, "The plumbing of land surface models: benchmarking model performance," *J. Hydrometeorol.*, vol. 16, no. June 2015, pp. 1425–1442, 2015.

[101] P. Willems, "Parsimonious rainfall-runoff model construction supported by time series processing and validation of hydrological extremes - Part 1: Step-wise model-structure identification and calibration approach," *J. Hydrol.*, vol. 510, pp. 578–590, 2014.

[102] J. Thielen, J. Bartholmes, M.-H. Ramos, and a. de Roo, "The European Flood Alert System &ndash; Part 1: Concept and development," *Hydrol. Earth Syst. Sci. Discuss.*, vol. 5, pp. 257–287, 2008.

[103] P. Lin, M. A. Rajib, Z. L. Yang, M. Somos-Valenzuela, V. Merwade, D. R. Maidment, Y. Wang, and L. Chen, "Spatiotemporal Evaluation of Simulated Evapotranspiration and Streamflow over Texas Using the WRF-Hydro-RAPID Modeling Framework," *J. Am. Water Resour. Assoc.*, vol. 54, no. 1, pp. 40–54, 2018.

[104] I. A. Thomas, P. Jordan, O. Shine, O. Fenton, P. E. Mellander, P. Dunlop, and P. N. C. Murphy, "Defining optimal DEM resolutions and point densities for modelling hydrologically sensitive areas in agricultural catchments dominated by microtopography," *Int. J. Appl. Earth Obs. Geoinf.*, vol. 54, pp. 38–52, 2017.

[105] J. Koch, T. Cornelissen, Z. Fang, H. Bogena, B. Diekkrüger, S. Kollet, and S. Stisen, "Inter-comparison of three distributed hydrological models with respect to seasonal variability of soil moisture patterns at a small forested catchment," *J. Hydrol.*, vol. 533, pp. 234–249, 2016.

[106] N. W. Chaney, J. K. Roundy, J. E. Herrera-Estrada, and E. F. Wood, "High-resolution modeling of the spatial heterogeneity of soil moisture: Applications in network design," *Water Resour. Res.*, pp. 1–20, 2014.

[107] E. F. Wood, J. K. Roundy, T. J. Troy, L. P. H. van Beek, M. F. P. Bierkens, E. Blyth, A. de Roo, P. Döll, M. Ek, J. Famiglietti, D. Gochis, N. van de Giesen, P. Houser, P. R. Jaffé, S. Kollet, B. Lehner, D. P. Lettenmaier, C. Peters-Lidard, M. Sivapalan, J. Sheffield, A. Wade, and P. Whitehead, "Hyperresolution global land surface modeling: Meeting a grand challenge for monitoring Earth's terrestrial water," *Water Resour. Res.*, vol. 47, no. 5, p. n/a-n/a, May 2011.

[108] M. A. Souffront Alcantara, S. Crawley, M. J. Stealey, E. J. Nelson, D. P. Ames, and N. L. Jones, "Open Water Data Solutions for Accessing the National Water Model," *Open Water J.*, vol. 4, no. 1, p. 3, 2017.

[109] C. M. DeChant and H. Moradkhani, "Hydrologic prediction and uncertainty quantification," *Handb. Eng. Hydrol. Model. Clim. Chang. Var. Saeid Eslamian (Editor). CRC Press Taylor Fr. Group, Boca Raton, Florida*, pp. 387–414, 2014.

[110] G. W. Feyereisen, T. C. Strickland, D. D. Bosch, D. G. Sullivan, D. G. S. G. W. Feyereisen, T. C. Strickland, D. D. Bosch, G. W. Feyereisen, T. C. Strickland, D. D. Bosch, and D. G. Sullivan, "Evaluation of SWAT manual calibration and input parameter sensitivity in the Little River watershed," *Trans. ASABE*, vol. 50, no. 3, pp. 843–855, 2007.

[111] E. L. Peck, *Catchment modeling and initial parameter estimation for the National Weather Service river forecast system*. Citeseer, 1976.

[112] J. C. Refsgaard and B. Storm, "Construction, calibration and validation of hydrological models," in *Distributed hydrological modelling*, Springer, 1990, pp. 41–54.

[113] T. Vansteenkiste, M. Tavakoli, N. Van Steenbergen, F. De Smedt, O. Batelaan, F. Pereira, and P. Willems, "Intercomparison of five lumped and distributed models for catchment runoff and extreme flow simulation," *J. Hydrol.*, vol. 511, pp. 335–349, 2014.

[114] A. M. Lumb, R. B. McCammon, and J. L. Kittle Jr, "Users Manual for an Expert System (HSPEXP) for Calibration of the Hydrological Simulation Program-FORTRAN," *Water-Resources Investig. Rep.*, pp. 94–4168, 1994.

[115] W. Chen and K. W. Chau, "Intelligent manipulation and calibration of parameters for hydrological models," *Int. J. Environ. Pollut.*, vol. 28, no. 3/4, p. 432, 2006.

[116] K. C. Abbaspour, E. Rouholahnejad, S. Vaghefi, R. Srinivasan, H. Yang, and B. Kløve, "A continental-scale hydrology and water quality model for Europe: Calibration and uncertainty of a high-resolution large-scale SWAT model," *J. Hydrol.*, vol. 524, pp. 733–752, 2015.

[117] X. Song, J. Zhang, C. Zhan, Y. Xuan, M. Ye, and C. Xu, "Global sensitivity analysis in hydrological modeling: Review of concepts, methods, theoretical framework, and applications," *J. Hydrol.*, vol. 523, no. 225, pp. 739–757, 2015.

[118] F. Pianosi, K. Beven, J. Freer, J. W. Hall, J. Rougier, D. B. Stephenson, and T. Wagener, "Sensitivity analysis of environmental models: A systematic review with practical workflow," *Environ. Model. Softw.*, vol. 79, pp. 214–232, 2016.

[119] V. K. Gupta and H. Sorooshian, "The automatic calibration of conceptual catchment models using derivative based optimization algorithms," *Water Resour. Res.*, vol. 21, no. 4, pp. 473–485, 1985.

[120] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer Journal*, vol. 7, no. 4. pp. 308–313, 1965.

[121] D. E. Goldberg and J. H. Holland, "Genetic Algorithms and Machine Learning," *Mach. Learn.*, vol. 3, no. 2, pp. 95–99, 1988.

[122] Q. J. Wang, "The Genetic Algorithm and Its Application to Calibrating Conceptual Rainfall-Runoff Models," *Water Resour. Res.*, vol. 27, no. 9, pp. 2467–2471, 1991.

[123] R. Storn and K. Price, “Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces,” *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997.

[124] R. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” *MHS’95. Proc. Sixth Int. Symp. Micro Mach. Hum. Sci.*, pp. 39–43, 1995.

[125] Y. Jiang, C. Liu, C. Huang, and X. Wu, “Improved particle swarm algorithm for hydrological parameter optimization,” *Appl. Math. Comput.*, vol. 217, no. 7, pp. 3207–3215, 2010.

[126] Q. Duan, S. Sorooshian, and V. Gupta, “Effective and efficient global optimization for conceptual rainfall-runoff models,” *Water Resour. Res.*, vol. 28, no. 4, pp. 1015–1031, 1992.

[127] Q. Duan, S. Sorooshian, and V. K. Gupta, “Optimal use of the SCE-UA global optimization method for calibrating watershed models,” *J. Hydrol.*, vol. 158, no. 3–4, pp. 265–284, Jun. 1994.

[128] K. Beven, “Prophecy, reality and uncertainty in distributed hydrological modelling,” *Adv. Water Resour.*, vol. 16, no. 1, pp. 41–51, 1993.

[129] K. Beven and J. Freer, “Hydrology Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems using the GLUE methodology,” *lournal Hydrol.*, vol. 249, pp. 11–29, 2001.

[130] K. C. Abbaspour, J. Yang, I. Maximov, R. Siber, K. Bogner, J. Mieleitner, J. Zobrist, and R. Srinivasan, “Modelling hydrology and water quality in the pre-alpine/alpine Thur watershed using SWAT,” *J. Hydrol.*, vol. 333, no. 2–4, pp. 413–430, 2007.

[131] J. Götzinger and A. Bárdossy, “Generic error model for calibration and uncertainty estimation of hydrological models,” *Water Resour. Res.*, vol. 44, no. 12, pp. 1–18, 2008.

[132] J. Yang, P. Reichert, K. C. Abbaspour, J. Xia, and H. Yang, “Comparing uncertainty analysis techniques for a SWAT application to the Chaohe Basin in China,” *J. Hydrol.*, vol. 358, no. 1–2, pp. 1–23, 2008.

[133] K. Beven and A. M. Binley, “The future of distributed models: model calibration and uncertainty estimation,” *Hydrol. Process.*, vol. 6, no. May 1991, pp. 279–298, 1992.

[134] K. Beven and A. Binley, “GLUE: 20 years on,” *Hydrol. Process.*, vol. 28, no. 24, pp. 5897–5918, 2014.

[135] G. Kuczera and E. Parent, “Monte Carlo assessment of parameter uncertainty in conceptual catchment models: The Metropolis algorithm,” *J. Hydrol.*, vol. 211, no. 1–4, pp. 69–85, 1998.

[136] J. A. Vrugt, C. J. F. ter Braak, C. G. H. Diks, and G. Schoups, “Hydrologic data assimilation using particle Markov chain Monte Carlo simulation: Theory, concepts and applications,” *Adv. Water Resour.*, vol. 51, no. 0, pp. 457–478, 2013.

[137] J. A. Vrugt, C. J. F. ter Braak, M. P. Clark, J. M. Hyman, and B. A. Robinson, “Treatment of input uncertainty in hydrologic modeling: Doing hydrology backward with Markov chain Monte Carlo simulation,” *Water Resour. Res.*, vol. 44, no. 12, 2008.

[138] J. A. Vrugt, C. J. F. ter Braak, C. G. H. Diks, B. A. Robinson, J. M. Hyman, and D. Higdon, “Accelerating Markov Chain Monte Carlo Simulation by Differential Evolution with Self-Adaptive Randomized Subspace Sampling,” *Int. J. Nonlinear Sci. Numer. Simul.*, vol. 10, no. 3, 2009.

[139] E. Laloy and J. A. Vrugt, “High-dimensional posterior exploration of hydrologic models using multiple-try DREAM (ZS) and high-performance computing,” *Water Resour. Res.*, vol. 48, no. 1, 2012.

[140] J. A. Vrugt, “Markov chain Monte Carlo simulation using the DREAM software package: Theory, concepts, and MATLAB implementation,” *Environ. Model. Softw.*, vol. 75, pp. 273–316, 2016.

[141] K. C. Abbaspour, C. A. Johnson, and M. T. van Genuchten, “Estimating Uncertain Flow and Transport Parameters Using a Sequential Uncertainty Fitting Procedure,” *Vadose Zo. J.*, vol. 3, no. 4, p. 1340, 2004.

[142] K. C. Abbaspour, M. T. Van Genuchten, R. Schulin, and E. Schläppi, “A sequential uncertainty domain inverse procedure for estimating subsurface flow and transport parameters,” *Water Resour. Res.*, vol. 33, no. 8, pp. 1879–1892, 1997.

[143] Y. Jiang, C. Liu, X. Li, L. Liu, and H. Wang, “Rainfall-runoff modeling, parameter estimation and sensitivity analysis in a semiarid catchment,” *Environ. Model. Softw.*, vol. 67, pp. 72–88, 2015.

[144] Q. Wu, S. Liu, Y. Cai, X. Li, and Y. Jiang, “Improvement of hydrological model calibration by selecting multiple parameter ranges,” *Hydrol. Earth Syst. Sci.*, vol. 21, no. 1, pp. 393–407, 2017.

[145] B. A. Tolson and C. A. Shoemaker, “Dynamically dimensioned search algorithm for computationally efficient watershed model calibration,” *Water Resour. Res.*, vol. 43, no. 1, pp. 1–16, 2007.

[146] W. Chu, X. Gao, and S. Sorooshian, “Improving the shuffled complex evolution scheme for optimization of complex nonlinear hydrological systems: Application to the calibration of the Sacramento soil-moisture accounting model,” *Water Resour. Res.*, vol. 46, no. 9, pp. 1–12, 2010.

[147] P. Acharjee and S. K. Goswami, “Expert algorithm based on adaptive particle swarm optimization for power flow analysis,” *Expert Syst. Appl.*, vol. 36, no. 3 PART 1, pp. 5151–5156, 2009.

[148] S. Muthukaruppan and M. J. Er, “A hybrid particle swarm optimization based fuzzy expert system for the diagnosis of coronary artery disease,” *Expert Syst. Appl.*, vol. 39, no. 14, pp. 11657–11665, 2012.

[149] J. G. Saldarriaga, S. Takahashi, F. Hernández, D. M. Díaz, and S. Ochoa, “An Energy Methodology for the Design of Water Distribution Systems,” in *World Environmental and Water Resources Congress 2010*, 2010, vol. 1, pp. 4303–4313.

[150] M. Sadegh, J. A. Vrugt, H. V Gupta, and C. Xu, “The soil water characteristic as new class of closed-form parametric expressions for the flow duration curve,” *J. Hydrol.*, no. JANUARY, pp. 1–53, 2016.

[151] L. M. Parada, J. P. Fram, and X. Liang, “Multi-Resolution Calibration Methodology for Hydrologic Models: Application to a Sub-Humid Catchment,” in *Calibration of Watershed Models*, American Geophysical Union, 2003, pp. 197–211.

[152] D. P. Boyle, H. V. Gupta, and S. Sorooshian, “Toward improved calibration of hydrologic models: Combining the strengths of manual and automatic methods,” *Water Resour. Res.*, vol. 36, no. 12, pp. 3663–3674, 2000.

[153] A. Efstratiadis and D. Koutsoyiannis, “Fitting Hydrological Models on Multiple Responses Using the Multiobjective Evolutionary Annealing-Simplex Approach,” *Pract. Hydroinformatics*, pp. 259–273, 2008.

[154] J. Chen, B. Xin, Z. Peng, L. Dou, and J. Zhang, “Optimal contraction theorem for exploration-exploitation tradeoff in search and optimization,” *IEEE Trans. Syst. Man, Cybern. Part ASystems Humans*, vol. 39, no. 3, pp. 680–691, 2009.

[155] Y. Leung, Y. Gao, and Z. Ben Xu, “Degree of population diversity - A perspective on premature convergence in genetic algorithms and its Markov chain analysis,” *IEEE Trans. Neural Networks*, vol. 8, no. 5, pp. 1165–1176, 1997.

[156] K. Deb, “Multi-objective Optimization,” in *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*, E. K. Burke and G. Kendall, Eds. Springer US, 2014, pp. 403–449.

[157] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Courier Corporation, 1998.

[158] M. Dorigo, M. Birattari, and T. Stutzle, “Ant colony optimization,” *IEEE Comput. Intell. Mag.*, vol. 1, no. November, 2006.

[159] G. Evin, M. Thyer, D. Kavetski, D. McInerney, G. Kuczera, E. Guillaume, T. Mark, K. Dmitri, M. David, K. George, G. Evin, M. Thyer, D. Kavetski, D. McInerney, G. Kuczera,

E. Guillaume, T. Mark, K. Dmitri, M. David, and K. George, “Comparison of joint versus postprocessor approaches for hydrological uncertainty estimation accounting for error autocorrelation and heteroscedasticity,” *Water Resour. Res.*, vol. 50, no. 3, pp. 2350–2375, Feb. 2014.

[160] G. Schoups and J. A. Vrugt, “A formal likelihood function for parameter and predictive inference of hydrologic models with correlated, heteroscedastic, and non-Gaussian errors,” *Water Resour. Res.*, vol. 46, no. 10, pp. 1–17, 2010.

[161] X. Liang, E. F. Wood, and D. P. Lettenmaier, “Surface soil moisture parameterization of the VIC-2L model: Evaluation and modification,” *Glob. Planet. Change*, vol. 13, no. 1, pp. 195–206, 1996.

[162] X. Liang, D. P. Lettenmaier, and E. F. Wood, “One-dimensional statistical dynamic representation of subgrid spatial variability of precipitation in the two-layer variable infiltration capacity model,” *J. Geophys. Res. Atmos.*, vol. 101, no. D16, pp. 21403–21422, 1996.

[163] X. Liang and Z. Xie, “A new surface runoff parameterization with subgrid-scale soil heterogeneity for land surface models,” *Adv. Water Resour.*, vol. 24, no. 9, pp. 1173–1193, 2001.

[164] E. Du, T. E. Link, J. A. Gravelle, and J. A. Hubbart, “Validation and sensitivity test of the distributed hydrology soil-vegetation model (DHSVM) in a forested mountain watershed,” *Hydrol. Process.*, vol. 28, no. 26, pp. 6196–6210, 2014.

[165] R. Sun, H. Yuan, and X. Liu, “Effect of heteroscedasticity treatment in residual error models on model calibration and prediction uncertainty estimation,” *J. Hydrol.*, vol. 554, no. October, pp. 680–692, 2017.

[166] M. A. Rajib, V. Merwade, and Z. Yu, “Multi-objective calibration of a hydrologic model using spatially distributed remotely sensed/in-situ soil moisture,” *J. Hydrol.*, vol. 536, pp. 192–207, 2016.

[167] N. Wanders, M. F. P. Bierkens, S. M. de Jong, A. de Roo, and D. Karssenberg, “The benefits of using remotely sensed soil moisture in parameter identification of large-scale hydrological models,” *Water Resour. Res.*, vol. 50, no. 8, pp. 6874–6891, 2014.

[168] F. Hernández and X. Liang, “Hybridizing Bayesian and variational data assimilation for high-resolution hydrologic forecasting,” *Hydrol. Earth Syst. Sci.*, vol. 22, no. 11, pp. 5759–5779, Nov. 2018.

[169] G. Evensen, *Data assimilation: the ensemble Kalman filter*. Springer Science & Business Media, 2009.

[170] A. Smith, A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice*. Springer Science & Business Media, 2013.

[171] Y. Zhu, Z. Toth, R. Wobus, D. Richardson, and K. Mylne, “The economic value of ensemble-based weather forecasts,” *Bull. Am. Meteorol. Soc.*, vol. 83, no. 1, pp. 73–83, 2002.

[172] J. S. Verkade and M. G. F. Werner, “Estimating the benefits of single value and probability forecasting for flood warning,” *Hydrol. Earth Syst. Sci.*, vol. 15, no. 12, pp. 3751–3765, 2011.

[173] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson, “Obstacles to High-Dimensional Particle Filtering,” *Mon. Weather Rev.*, vol. 136, no. 12, pp. 4629–4640, 2008.

[174] P. J. van Leeuwen, “Nonlinear Data Assimilation for high-dimensional systems,” in *Nonlinear Data Assimilation*, J. P. Van Leeuwen, Y. Cheng, and S. Reich, Eds. Springer International Publishing, 2015, pp. 1–73.

[175] R. H. Reichle, D. B. McLaughlin, and D. Entekhabi, “Variational data assimilation of microwave radiobrightness observations for land surface hydrology applications,” *Geosci. Remote Sensing, IEEE Trans.*, vol. 39, no. 8, pp. 1708–1718, Aug. 2001.

[176] M. Ghil and P. Malanotte-Rizzoli, “Data assimilation in meteorology and oceanography,” *Adv. Geophys*, vol. 33, pp. 141–266, 1991.

[177] Y. Liu and H. V. Gupta, “Uncertainty in hydrologic modeling: Toward an integrated data assimilation framework,” *Water Resour. Res.*, vol. 43, no. 7, pp. 1–18, 2007.

[178] R. M. Errico, “What Is an Adjoint Model?,” *Bull. Am. Meteorol. Soc.*, vol. 78, no. 11, pp. 2577–2591, 1997.

[179] R. N. Bannister, “A review of operational methods of variational and ensemble-variational data assimilation,” *Q. J. R. Meteorol. Soc.*, vol. 29, no. January, pp. 1–29, 2016.

[180] F. Zhang, M. Zhang, and J. Hansen, “Coupling ensemble Kalman filter with four-dimensional variational data assimilation,” *Adv. Atmos. Sci.*, vol. 26, no. 1, pp. 1–8, 2009.

[181] G. Dumedah and P. Coulibaly, “Evolutionary assimilation of streamflow in distributed hydrologic modeling using in-situ soil moisture data,” *Adv. Water Resour.*, vol. 53, pp. 231–241, 2013.

[182] S. Park, J. P. Hwang, E. Kim, and H. J. Kang, “A new evolutionary particle filter for the prevention of sample impoverishment,” *IEEE Trans. Evol. Comput.*, vol. 13, no. 4, pp. 801–809, 2009.

[183] M. Buehner, P. L. Houtekamer, C. Charette, H. L. Mitchell, and B. He, “Intercomparison of Variational Data Assimilation and the Ensemble Kalman Filter for Global Deterministic NWP. Part II: One-Month Experiments with Real Observations,” *Mon. Weather Rev.*, vol. 138, no. 5, pp. 1567–1586, 2010.

[184] G. Desroziers, J. T. Camino, and L. Berre, “4DEnVar: Link with 4D state formulation of variational assimilation and different possible implementations,” *Q. J. R. Meteorol. Soc.*, vol. 140, no. 684, pp. 2097–2110, 2014.

[185] A. C. Lorenc, N. E. Bowler, A. M. Clayton, S. R. Pring, and D. Fairbairn, “Comparison of Hybrid-4DEnVar and Hybrid-4DVar Data Assimilation Methods for Global NWP,” *Mon. Weather Rev.*, vol. 143, no. 1, pp. 212–229, 2015.

[186] Y. Trémolet, “Accounting for an imperfect model in 4D-Var,” *Q. J. R. Meteorol. Soc.*, vol. 132, no. 621, pp. 2483–2504, 2006.

[187] L. Ning, F. P. Carli, A. M. Ebtehaj, E. Foufoula-Georgiou, and T. T. Georgiou, “Coping with model error in variational data assimilation using optimal mass transport,” *Water Resour. Res.*, vol. 50, no. 7, pp. 5817–5830, 2014.

[188] G. Evensen and P. J. van Leeuwen, “An ensemble Kalman smoother for nonlinear dynamics,” *Mon. Weather Rev.*, vol. 128, no. 6, pp. 1852–1867, 2000.

[189] S. J. Noh, Y. Tachikawa, M. Shiiba, and S. Kim, “Applying sequential Monte Carlo methods into a distributed hydrologic model: Lagged particle filtering approach with regularization,” *Hydrol. Earth Syst. Sci.*, vol. 15, no. 10, pp. 3237–3251, 2011.

[190] N. J. Gordon, D. J. Salmond, and A. F. M. Smith, “Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” *IEE Proc. F Radar Signal Process.*, vol. 140, no. 2, pp. 107–113, 1993.

[191] P. J. van Leeuwen, “Particle Filtering in Geophysical Systems,” *Mon. Weather Rev.*, vol. 137, no. 12, pp. 4089–4114, 2009.

[192] P. Gauthier, M. Tanguay, S. Laroche, S. Pellerin, and J. Morneau, “Extension of 3DVAR to 4DVAR: Implementation of 4DVAR at the Meteorological Service of Canada,” *Mon. Weather Rev.*, vol. 135, no. 6, pp. 2339–2354, 2007.

[193] F. Rawlins, S. P. Ballard, K. J. Bovis, A. M. Clayton, D. Li, G. W. Inverarity, A. C. Lorenc, and T. J. Payne, “The Met Office global four-dimensional variational data assimilation scheme,” *Q. J. R. Meteorol. Soc.*, vol. 133, no. 623, pp. 347–362, Jan. 2007.

[194] S.-C. Yang, M. Corazza, A. Carrassi, E. Kalnay, and T. Miyoshi, “Comparison of Local Ensemble Transform Kalman Filter, 3DVAR, and 4DVAR in a Quasigeostrophic Model,” *Mon. Weather Rev.*, vol. 137, no. 2, pp. 693–709, 2009.

[195] M. Fisher, “Background error covariance modelling,” *Semin. Recent Dev. Data Assim. …*, pp. 45–63, 2003.

[196] M. P. Clark, D. E. Rupp, R. a. Woods, X. Zheng, R. P. Ibbitt, A. G. Slater, J. Schmidt, and M. J. Uddstrom, “Hydrological data assimilation with the ensemble Kalman filter: Use of streamflow observations to update states in a distributed hydrological model,” *Adv. Water Resour.*, vol. 31, no. 10, pp. 1309–1324, 2008.

[197] P. R. Houser, W. J. Shuttleworth, J. S. Famiglietti, H. V Gupta, K. H. Syed, and D. C. Goodrich, "Integration of soil moisture remote sensing and hydrologic modeling using data assimilation," *Water Resour. Res.*, vol. 34, no. 12, pp. 3405–3420, 1998.

[198] V. Maggioni and P. R. Houser, "Soil Moisture Data Assimilation," in *Data Assimilation for Atmospheric, Oceanic and Hydrologic Applications (Vol. III)*, vol. III, S. K. Park and L. Xu, Eds. Cham: Springer International Publishing, 2017, pp. 195–217.

[199] K. M. Andreadis and D. P. Lettenmaier, "Assimilating remotely sensed snow observations into a macroscale hydrology model," *Adv. Water Resour.*, vol. 29, no. 6, pp. 872–886, 2006.

[200] M. West, "Mixture models, Monte Carlo, Bayesian updating, and dynamic models," *Comput. Sci. Stat.*, pp. 1–11, 1993.

[201] J. Carpenter, P. Clifford, and P. Fearnhead, "Improved particle filter for nonlinear problems," *IEE Proc. - Radar, Sonar Navig.*, vol. 146, no. 1, pp. 2–7, 1999.

[202] J. S. Liu and R. Chen, "Sequential Monte Carlo Methods for Dynamic Systems," *J. Am. Stat. Assoc.*, vol. 93, no. 443, pp. 1032–1044, 1998.

[203] C. Montzka, V. R. N. Pauwels, H.-J. H. Franssen, X. Han, and H. Vereecken, "Multivariate and Multiscale Data Assimilation in Terrestrial Systems: A Review," *Sensors*, vol. 12, no. 12, pp. 16291–16333, 2012.

[204] G. Dumedah, A. a. Berg, and M. Wineberg, "An Integrated Framework for a Joint Assimilation of Brightness Temperature and Soil Moisture Using the Nondominated Sorting Genetic Algorithm II," *J. Hydrometeorol.*, vol. 12, no. 6, pp. 1596–1609, Apr. 2011.

[205] X. Zhang, Y. Tian, R. Cheng, and Y. Jin, "An Efficient Approach to Nondominated Sorting for Evolutionary Multiobjective Optimization," *Evol. Comput. IEEE Trans.*, vol. 19, no. 2, pp. 201–213, 2015.

[206] M. P. Wand and M. C. Jones, *Kernel smoothing*. Crc Press, 1994.

[207] S. J. Sheather and M. C. Jones, "A Reliable Data-Based Bandwidth Selection Method for Kernel Density Estimation," *J. R. Stat. Soc. Ser. B*, vol. 53, no. 3, pp. 683–690, Jan. 1991.

[208] T. Duong and M. L. Hazelton, "Cross-validation bandwidth matrices for multivariate kernel density estimation," *Scand. J. Stat.*, vol. 32, pp. 485–506, 2005.

[209] G. R. Terrell and D. W. Scott, "Variable kernel density estimation," *Ann. Stat.*, vol. 20, no. 3, pp. 1236–1265, 1992.

[210] M. L. Hazelton, "Variable kernel density estimation," *Aust. N. Z. J. Stat.*, vol. 45, no. 3, pp. 271–284, Sep. 2003.

[211] A. Krishnamoorthy and D. Menon, “Matrix inversion using Cholesky decomposition,” *2013 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*. pp. 70–72, 2013.

[212] J. Friedman, T. Hastie, and R. Tibshirani, “Sparse inverse covariance estimation with the graphical lasso,” *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[213] P. C. Mahalanobis, “On the generalized distance in statistics,” *Proc. Natl. Inst. Sci.*, vol. 2, pp. 49–55, 1936.

[214] X. Liang and Z. Xie, “Important factors in land–atmosphere interactions: surface runoff generations and interactions between surface and groundwater,” *Glob. Planet. Change*, vol. 38, no. 1, pp. 101–114, 2003.

[215] J. Guo, X. Liang, and L. R. Leung, “A new multiscale flow network generation scheme for land surface models,” *Geophys. Res. Lett.*, vol. 31, no. 23, pp. 1–4, 2004.

[216] Z. Wen, X. Liang, and S. Yang, “A new multiscale routing framework and its evaluation for land surface modeling applications,” *Water Resour. Res.*, vol. 48, no. June, pp. 1–16, 2012.

[217] D. Lohmann, E. Rashke, B. Nijssen, and D. P. Lettenmaier, “Regional scale hydrology: I. Formulation of the VIC-2L model coupled to a routing model,” *Hydrol. Sci. J.*, vol. 43, no. 1, pp. 131–141, 1998.

[218] P. Krause, D. P. Boyle, and F. Bäse, “Comparison of different efficiency criteria for hydrological model assessment,” *Adv. Geosci.*, vol. 5, no. 89, pp. 89–97, 2005.

[219] D. T. Pham, “Stochastic methods for sequential data assimilation in strongly nonlinear systems,” *Mon. Weather Rev.*, vol. 129, no. 5, pp. 1194–1207, 2001.

[220] H. Moradkhani, C. M. DeChant, and S. Sorooshian, “Evolution of ensemble data assimilation for uncertainty quantification using the particle filter-Markov chain Monte Carlo method,” *Water Resour. Res.*, vol. 48, no. 12, pp. 1–13, 2012.

[221] J. Bröcker, “Evaluating raw ensembles with the continuous ranked probability score,” *Q. J. R. Meteorol. Soc.*, vol. 138, no. 667, pp. 1611–1617, 2012.

[222] D. C. Montgomery, G. C. Runger, and N. F. Hubele, *Engineering statistics*. John Wiley & Sons, 2009.

[223] G. Karafotias, M. Hoogendoorn, and A. E. Eiben, “Parameter Control in Evolutionary Algorithms: Trends and Challenges,” *IEEE Trans. Evol. Comput.*, vol. to appear, no. 2, pp. 167–187, 2014.

[224] J. Y. Li, A. Kokkinaki, H. Ghorbanidehno, E. F. Darve, and P. K. Kitanidis, “The compressed state Kalman filter for nonlinear state estimation: Application to large-scale reservoir monitoring,” *Water Resour. Res.*, vol. 51, no. 12, pp. 9942–9963, Dec. 2015.

[225] H. Ghorbanidehno, A. Kokkinaki, J. Y. Li, E. Darve, and P. K. Kitanidis, “Real-time data assimilation for large-scale systems: The spectral Kalman filter,” *Adv. Water Resour.*, vol. 86, pp. 260–272, 2015.

[226] R. N. Bannister, “A review of forecast error covariance statistics in atmospheric variational data assimilation. II: Modelling the forecast error covariance statistics,” *Q. J. R. Meteorol. Soc.*, vol. 134, no. 637, pp. 1971–1996, 2008.

[227] H. Ajami, U. Khan, N. K. Tuteja, and A. Sharma, “Development of a computationally efficient semi-distributed hydrologic modeling application for soil moisture, lateral flow and runoff simulation,” *Environ. Model. Softw.*, vol. 85, pp. 319–331, 2016.

[228] A. K. Jain, “Data clustering: 50 years beyond K-means,” *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[229] N. W. Chaney, P. Metcalfe, and E. F. Wood, “HydroBlocks: a field-scale resolving land surface model for application over continental extents,” *Hydrol. Process.*, vol. 30, no. 20, pp. 3543–3559, 2016.

[230] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data clustering: a review,” *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.

[231] E. H. Ruspini, “A new approach to clustering,” *Inf. Control*, vol. 15, no. 1, pp. 22–32, 1969.

[232] P. Bauer, A. Thorpe, and G. Brunet, “The quiet revolution of numerical weather prediction,” *Nature*, vol. 525, no. 7567, pp. 47–55, 2015.

[233] C. Robert and G. Casella, *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

[234] H. Moradkhani, G. Nearing, P. Abbaszadeh, and S. Pathiraja, “Fundamentals of Data Assimilation and Theoretical Advances,” *Handb. Hydrometeorol. Ensemble Forecast.*, pp. 1–26, 2018.

[235] A. Carrassi, M. Bocquet, L. Bertino, and G. Evensen, “Data assimilation in the geosciences: An overview of methods, issues, and perspectives,” *Wiley Interdiscip. Rev. Clim. Chang.*, vol. 9, no. 5, p. e535, 2018.

[236] S. J. Fletcher, “Applications of Data Assimilation in the Geosciences,” in *Data Assimilation for the Geosciences*, S. J. B. T.-D. A. for the G. Fletcher, Ed. Elsevier, 2017, pp. 887–916.

[237] R. N. Bannister, “A review of forecast error covariance statistics in atmospheric variational data assimilation. I: Characteristics and measurements of forecast error covariances,” *Q. J. R. Meteorol. Soc.*, vol. 134, no. 637, pp. 1951–1970, Oct. 2008.

[238] M. Yuan and Y. Lin, “Model selection and estimation in the Gaussian graphical model,” *Biometrika*, vol. 94, no. 1, pp. 19–35, 2007.

[239] N. Meinshausen and P. Bühlmann, “High-dimensional graphs and variable selection with the Lasso,” *Ann. Stat.*, vol. 34, no. 3, pp. 1436–1462, 2006.

[240] S. Sojoudi, “Equivalence of graphical lasso and thresholding for sparse graphs,” *J. Mach. Learn. Res.*, vol. 17, no. 115, pp. 1–21, 2016.

[241] A. Ali, K. Khare, S.-Y. Oh, and B. Rajaratnam, “Generalized Pseudolikelihood Methods for Inverse Covariance Estimation,” no. 2007, pp. 1–41, 2016.

[242] K. Khare, S. Y. Oh, and B. Rajaratnam, “A convex pseudolikelihood framework for high dimensional partial correlation estimation with convergence guarantees,” *J. R. Stat. Soc. Ser. B Stat. Methodol.*, vol. 77, no. 4, pp. 803–825, 2015.

[243] J. Peng, P. Wang, N. Zhou, and J. Zhu, “Partial correlation estimation by joint sparse regression models,” *J. Am. Stat. Assoc.*, vol. 104, no. 486, pp. 735–746, 2009.

[244] J. Friedman, T. Hastie, and R. Tibshirani, “Applications of the lasso and grouped lasso to the estimation of sparse graphical models,” *Tech. Rep.*, pp. 1–22, 2010.

[245] G. V. Rocha, P. Zhao, and B. Yu, “A path following algorithm for Sparse Pseudo-Likelihood Inverse Covariance Estimation (SPLICE),” *Spring*, no. 60628102, p. 33, 2008.

[246] T. Wang, Z. Ren, Y. Ding, Z. Fang, Z. Sun, M. L. MacDonald, R. A. Sweet, J. Wang, and W. Chen, “FastGGM: An Efficient Algorithm for the Inference of Gaussian Graphical Model in Biological Networks,” *PLoS Comput. Biol.*, vol. 12, no. 2, pp. 1–16, 2016.

[247] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar, “QUIC: Quadratic Approximation for Sparse Inverse Covariance Estimation,” *J. Mach. Learn. Res.*, vol. 15, pp. 2911–2947, 2014.

[248] Z. Ren, T. Sun, C.-H. Zhang, and H. H. Zhou, “Asymptotic normality and optimalities in estimation of large Gaussian graphical models,” *Ann. Stat.*, vol. 43, no. 3, pp. 991–1026, 2015.

[249] O. Dalal and B. Rajaratnam, “Sparse Gaussian graphical model estimation via alternating minimization,” *Biometrika*, vol. 104, no. 2, pp. 379–395, 2017.

[250] G. G. R. Leday and S. Richardson, “Fast Bayesian inference in large Gaussian graphical models,” *arXiv Prepr. arXiv1803.08155*, pp. 1–26, 2018.

[251] A. Mohammadi and E. C. Wit, “Bayesian structure learning in sparse Gaussian graphical models,” *Bayesian Anal.*, vol. 10, no. 1, pp. 109–138, 2015.

[252] A. Mohammadi, F. Abegaz, E. van den Heuvel, and E. C. Wit, “Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models,” *J. R. Stat. Soc. Ser. C (Applied Stat.*, vol. 66, no. 3, pp. 629–645, 2017.

[253] J. Schäfer and K. Strimmer, “A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics,” *Stat. Appl. Genet. Mol. Biol.*, vol. 4, no. 1, 2005.

[254] D. M. Witten, J. H. Friedman, and N. Simon, “New insights and faster computations for the graphical lasso,” *J. Comput. Graph. Stat.*, vol. 20, no. 4, pp. 892–900, 2011.

[255] R. Mazumder and T. Hastie, “Exact covariance thresholding into connected components for large-scale Graphical Lasso,” vol. 13, pp. 723–736, 2011.

[256] T. Pavlenko, A. Björkström, and A. Tillander, “Covariance structure approximation via gLasso in high-dimensional supervised classification,” *J. Appl. Stat.*, vol. 39, no. 8, pp. 1643–1666, 2012.

[257] B. M. Marlin and K. P. Murphy, “Sparse Gaussian graphical models with unknown block structure,” *Proc. 26th Annu. Int. Conf. Mach. Learn. - ICML ’09*, pp. 1–8, 2009.

[258] S. Celik, B. A. Logsdon, and L. Su-In, “Efficient Dimensionality Reduction for High-Dimensional Network Estimation,” *Proc. 31st Int. Conf. Mach. Learn.*, vol. 32, pp. 1953–1961, 2014.

[259] K. M. Tan, D. Witten, and A. Shojaie, “The cluster graphical lasso for improved estimation of Gaussian graphical models,” *Comput. Stat. Data Anal.*, vol. 85, pp. 23–36, 2015.

[260] E. Devijver and M. Gallopin, “Block-diagonal covariance selection for high-dimensional Gaussian graphical models,” *J. Am. Stat. Assoc.*, vol. 113, no. 521, pp. 306–314, 2018.

[261] J. S. Yedidia, W. T. Freeman, and Y. Weiss, “Understanding belief propagation and its generalizations,” *Explor. Artif. Intell. new Millenn.*, vol. 8, pp. 236–239, 2003.

[262] Y. Weiss and W. T. Freeman, “Correctness of belief propagation in Gaussian graphical models of arbitrary topology.,” *Neural Comput.*, vol. 13, pp. 2173–2200, 2001.

[263] D. M. Malioutov, J. K. Johnson, and A. S. Willsky, “Walk-sums and belief propagation in Gaussian graphical models,” *J. Mach. Learn. Res.*, vol. 7, no. 7, pp. 2031–2064, 2006.

[264] J. Hosseini and S.-I. Lee, “Learning Sparse Gaussian Graphical Models with Overlapping Blocks,” *Neural Inf. Process. Syst.*, no. Nips, pp. 1–9, 2016.

[265] D. Koller and N. Friedman, *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

[266] J. D. Loudin and H. E. Miettinen, “A multivariate method for comparing n-dimensional distributions,” in *Proceedings of the Conference on Statistical Problems in Particle Physics, Astrophysics and Cosmology (PHYSTAT)*, 2003, no. 2, pp. 207–210.

[267] P. R. Rosenbaum, “An exact distribution-free test comparing two multivariate distributions based on adjacency,” *J. R. Stat. Soc. Ser. B (Statistical Methodol.*, vol. 67, no. 4, pp. 515–530, 2005.

[268] M. Roederer, W. Moore, A. Treister, R. R. Hardy, and L. A. Herzenberg, “Probability binning comparison: a metric for quantitating multivariate distribution differences,” *Cytom. J. Int. Soc. Anal. Cytol.*, vol. 45, no. 1, pp. 47–55, 2001.

[269] N. Fachada, V. V Lopes, R. C. Martins, and A. C. Rosa, “Model-independent comparison of simulation output,” *Simul. Model. Pract. Theory*, vol. 72, pp. 131–149, 2017.

[270] J. W. Mueller and T. Jaakkola, “Principal differences analysis: Interpretable characterization of differences between distributions,” in *Advances in Neural Information Processing Systems*, 2015, pp. 1702–1710.

[271] H. Moradkhani, S. Sorooshian, H. V. Gupta, and P. R. Houser, “Dual state-parameter estimation of hydrological models using ensemble Kalman filter,” *Adv. Water Resour.*, vol. 28, pp. 135–147, 2005.

[272] H. Moradkhani, K.-L. K. Hsu, H. V Gupta, and S. Sorooshian, “Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter,” *Water Resour. Res.*, vol. 41, no. 5, pp. 1–17, 2005.

[273] X. Xie and D. Zhang, “Data assimilation for distributed hydrological catchment modeling via ensemble Kalman filter,” *Adv. Water Resour.*, vol. 33, no. 6, pp. 678–690, 2010.

[274] J. A. Vrugt, C. G. H. Diks, H. V. Gupta, W. Bouten, and J. M. Verstraten, “Improved treatment of uncertainty in hydrologic modeling: Combining the strengths of global optimization and data assimilation,” *Water Resour. Res.*, vol. 41, pp. 1–17, 2005.