

# Advancing model calibration and uncertainty analysis of SWAT models using cloud computing infrastructure: LCC-SWAT

Masood Zamani, Narayan Kumar Shrestha, Taimoor Akhtar, Trevor Boston and Prasad Daggupati

## ABSTRACT

Calibration and uncertainty analysis of a complex, over-parameterized environmental model such as the Soil and Water Assessment Tool (SWAT) requires thousands of simulation runs and multiple calibration iterations. A parallel calibration system is thus desired that can be deployed on cloud-based architectures for reducing calibration runtime. This paper presents a cloud-based calibration and uncertainty analysis system called LCC-SWAT that is designed for SWAT models. Two optimization techniques, sequential uncertainty fitting (SUFI-2) and dynamically dimensioned search (DDS), have been implemented in LCC-SWAT. Moreover, the cloud-based system has been deployed on the Southern Ontario Smart Computing Innovation Platform's (SOSCIP) Cloud Analytics platform for diagnostic assessment of parallel calibration runtime on both single-node and multi-node CPU architectures. Unlike other calibrations/uncertainty analysis systems developed on the cloud, this system is capable of generating a comprehensive set of statistical information automatically, which facilitates broader analyses of the performance of the SWAT models. Experimental results on SWAT models of different complexities showed that LCC-SWAT can reduce runtime significantly. The runtime reduction is more pronounced for more complex and computationally intensive models. However, the reported runtime efficiency is significantly higher for single node systems. Comparative experiments with DDS and SUFI-2 show that parallel DDS outperforms parallel SUFI-2 in terms of both parameter identifiability and reducing uncertainty in model simulations. LCC-SWAT is a flexible calibration system and other optimization algorithms and asynchronous parallelization strategies can be added to it in future.

**Key words** | cloud computing, DDS, optimization, SUFI-2, SWAT

## HIGHLIGHTS

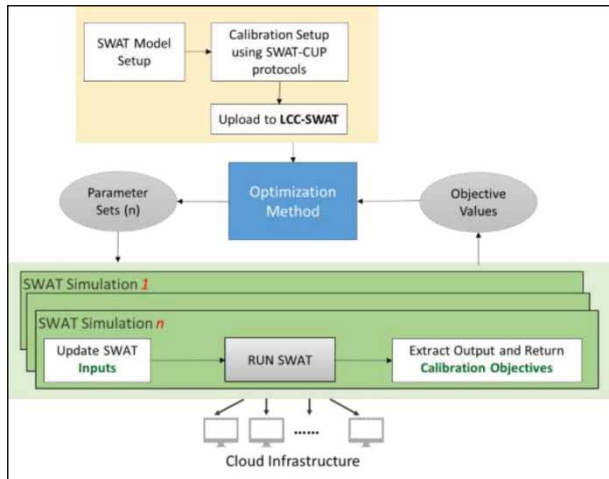
- LCC-SWAT: a cloud-based calibration and uncertainty analysis system for SWAT models.
- Two optimization techniques SUFI-2 and DDS have been implemented in the LCC-SWAT.
- LCC-SWAT is capable of generating a comprehensive set of statistical information.
- Experiments showed that LCC-SWAT can reduce runtime significantly.

**Masood Zamani**  
**Narayan Kumar Shrestha**  
**Taimoor Akhtar**  
**Prasad Daggupati** (corresponding author)  
School of Engineering,  
University of Guelph,  
50 Stone Road West, Guelph, ON,  
Canada  
N1G 2W1  
E-mail: [pdaggupa@uoguelph.ca](mailto:pdaggupa@uoguelph.ca)

**Masood Zamani**  
St. Michael's Hospital,  
University of Toronto,  
1 King's College Circle, Toronto, ON,  
Canada  
M5S 1A8

**Trevor Boston**  
Greenland International Consulting Ltd,  
120 Hume Street, Collingwood, ON,  
Canada  
L9Y 1V5

## GRAPHICAL ABSTRACT



## INTRODUCTION

Watershed models are widely used by water resources planners and managers in the decision-making process (Devia *et al.* 2015; Leta *et al.* 2015). With easier access to parallel computing power and the ready availability of higher-resolution observed datasets (including climate data, water quality data, etc.), the computational complexity of watershed models, and especially physically based distributed and semi-distributed watershed models, is increasing significantly (Yang *et al.* 2018). Moreover, complex physically based watershed models are typically characterized by a large number of parameters, and complex calibration objectives that are highly non-linear and multi-modal (Beven & Binley 1992; Abbaspour *et al.* 2004; Tolson & Shoemaker 2007). Hence, automatic parameter estimation of a complex watershed model is often hindered by high-dimensionality and multi-modality of the underlying calibration optimization problem (Nossent *et al.* 2011), and thus, the calibration process is computationally intensive (Ercan *et al.* 2014). The calibration challenge is exacerbated further by the need for understanding model uncertainty, especially for models that simulate water quality (e.g., maximum daily pollutants loads) (Shirmohammadi *et al.* 2006; Borah *et al.* 2019). The computationally intensive nature of complex watershed models, thus necessitates

optimization algorithms and frameworks that are computationally efficient and can use parallel computing resources (Ahmadisharaf *et al.* 2019).

The Soil and Water Assessment Tool (SWAT) is a highly popular watershed modeling tool (Arnold *et al.* 1998) that is widely used for the development of complex, highly parameterized and computationally expensive watershed models (Nossent *et al.* 2011; Borah *et al.* 2019). Many optimization algorithms have been developed in the past literature for addressing the computational challenge of calibrating SWAT and other complex watershed models (Tayfur 2017). For instance, Abbaspour *et al.* (2004) proposed Sequential Uncertainty Fitting (SUFI-2) to efficiently calibrate (within a few thousand simulations) complex SWAT models. Tolson & Shoemaker (2007) proposed the Dynamically Dimensioned Search (DDS) method to calibrate complex and high-dimensional (i.e., with many parameters) hydrologic and watershed models. Efficient Markov Chain Monte Carlo (MCMC) methods, e.g., the Shuffled Complex Evolution Metropolis (SCEM-UA) method (Vrugt *et al.* 2003) and the Differential Evolution Adaptive Metropolis (DREAM) algorithm (Vrugt *et al.* 2008; Vrugt 2016), and the Multiple-response Bayesian Calibration (MRBC) framework (Han & Zheng 2016) that quantify input and

parameter uncertainty during calibration have also been extensively applied to watershed problems. Given the inherent multi-objective nature of watershed model calibration (Gupta *et al.* 1998, 2009; Yapo *et al.* 1998), numerous multi-objective algorithms have also been proposed and used for watershed model calibration and uncertainty quantification. Some notable mentions are the ParaSol (van Griensven & Meixner 2007), the Borg Multiobjective Evolutionary Algorithm (Borg-MOEA) (Hadka & Reed 2013), the Pareto archived dynamically dimensioned search (PA-DDS) (Asadzadeh & Tolson 2013) and the Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb *et al.* 2002; Ercan & Goodall 2016). ParaSol (van Griensven & Meixner 2007) also applies thresholds on different objectives to filter/identify behavioral solutions (Beven & Binley 1992) and, subsequently, quantify model uncertainty.

The use of desktop/stand-alone computational resources is less effective and, in some cases, not feasible for automatic calibration and analysis of large-scale watershed models (especially distributed and semi-distributed models) with complex physical domains and multiple water resource issues such as water quality, droughts and floods (Abbaspour *et al.* 2007; Arnold *et al.* 2012; Gupta *et al.* 2012). Hence, parallel implementations of many calibration frameworks and algorithms have been introduced in the recent literature (Kan *et al.* 2019). Rouholahnejad *et al.* (2012) implemented a parallel version of the SUFI-2 algorithm that is widely used for calibration of SWAT models. SUFI-2 is part of SWAT-CUP (Abbaspour 2015), which is a popular stand-alone program for calibration of SWAT models. Ercan *et al.* (2014) implemented a parallel version of DDS for SWAT calibration with deployment on a windows-based cloud infrastructure. Joseph & Guillaume (2013) presented a parallel implementation of the DREAM algorithm that is specifically designed for parameter estimation and uncertainty quantification of SWAT watershed models. Bacu *et al.* (2011) developed grid-based architectural components for SWAT (gSWAT) with up to 60 worker nodes. The gSWAT with its inherent SUFI-2 algorithm was used in a fine resolution SWAT model of the Black Sea catchment (Rodila *et al.* 2015) in the scope of an EU/FP7 enviroGRIDS project (enviroGRIDS 2009) and in a large-scale Danube River Basin project (Gorgan *et al.* 2012;

Rodila *et al.* 2012). The gSWAT computing infrastructure was found to optimize the SWAT model when running in parallel (Bacu *et al.* 2017). Zhang *et al.* (2017) also parallelized the SWAT model itself (rather than the calibration framework) by simultaneously simulating output for each distributed model land unit (also called Hydrological Response Unit, HRU).

The above-mentioned parallel calibration frameworks clearly illustrate the value of parallel processing and high-performance computing in addressing the challenge of calibrating large and complex watershed models (Humphrey *et al.* 2012; Ercan *et al.* 2014; Astsatryan *et al.* 2016; Zhang *et al.* 2016). However, many prior studies do not adequately discuss two key aspects of parallel watershed model calibration, i.e., (i) compatibility of parallel algorithm implementations with different cloud platforms and (ii) the impact of cloud infrastructure on computing efficiency of parallel algorithms and frameworks.

In general, commercialized cloud computing platforms currently support either Windows or Linux operating systems, and in some cases, both operating systems are supported. Thus operating system compatibility is an important criterion in developing a cloud-based calibration platform. In Humphrey *et al.* (2012), a cloud-based calibration system for SWAT models was developed, based on Microsoft Windows Azure (Chappell 2009). In addition, a multi-component enterprise cloud service was developed and studied for watershed calibration, and the virtual machines (VMs) for the cloud platform were created using Hadoop and Openstack which are open-source software (Astsatryan *et al.* 2016; Zhang *et al.* 2016).

Hardware infrastructure of cloud platforms is also an important factor that can significantly impact speed-up and efficiency of frameworks where simulations are executed in parallel. O'Donncha *et al.* (2016) showed that parallel performance/efficiency of a fluid dynamics model varies significantly on single versus multi-node architectures. In their review of the parallel watershed calibration literature, Kan *et al.* (2019) note that prior work on understanding the effectiveness of parallel watershed model frameworks on different cloud platforms is very limited and needs to be explored more in future. Motivated by this need, we implemented a cloud-based watershed calibration and uncertainty analysis system under Linux Operating

System using SOSCIP's cloud analytic platform (SOSCIP 2019).

The watershed calibration system proposed in this study is called Linux-based Cloud Calibration system for SWAT (LCC-SWAT) and is specifically designed for complex watershed models developed using SWAT. LCC-SWAT includes two parallel optimization methods SUFI-2 (Abbaspour *et al.* 2004, 2007) and DDS (Tolson & Shoemaker 2007). The design of our cloud-based calibration system is compatible with the commonly used stand-alone SWAT-CUP system (Abbaspour 2015). We believe that this will encourage existing SWAT (and SWAT-CUP) users and modelers to test their models using LCC-SWAT. LCC-SWAT has been added to the Canadian Watershed Evaluation Tool (CANWET™) platform to provide efficient, automatic calibration and visualization of SWAT models.

This paper also includes a comprehensive comparison of parallel calibration results obtained from SUFI-2 and DDS on the SWAT model of the Grand River Basin (6,542 km<sup>2</sup>) in Ontario, Canada. To the best of our knowledge, parallel implementations of DDS and SUFI-2 have not been compared in the past. Moreover, the detailed runtime performance of LCC-SWAT is evaluated on different cloud architectures (i.e., single versus multi-node CPU systems), and by using three SWAT models of increasing complexities and sizes (19–215,918 km<sup>2</sup>). We believe that the availability of such a cloud-based system is an important contribution to watershed modeling software and to the future implementation of improved cloud-based calibration frameworks. In addition, unlike other cloud-based systems, LCC-SWAT automatically generates comprehensive statistical reports pertaining to the SWAT model calibration and uncertainty analysis which facilitates more comprehensive analyses of the calibration parameters and the overall model performance.

## BACKGROUND

### Soil and Water Assessment Tool

SWAT (Arnold *et al.* 1998) is a long-term continuous hydrologic and water quality model. It is one of the most widely used models in the hydro-environmental domain (Arnold *et al.* 2012). Being a semi-distributed and physically based

model, SWAT has a high number of parameters related to hydrology, erosion and sediment transport, nutrients, pesticides, fecal bacteria, among others (Leta *et al.* 2015), making it one of the more complex and over-parameterized hydro-environmental models (Nossent *et al.* 2011). For modeling purposes, SWAT divides a watershed into several sub-watersheds. A sub-watershed is further divided into Hydrological Response Units (HRUs) which are a unique combination of land-use, soil and slope. An HRU is the computation unit of the SWAT model (Arnold *et al.* 2011).

### Cloud computing infrastructure

The cloud computing infrastructure on which the new watershed calibration system was deployed and tested and was created by the Southern Ontario Smart Computing Innovation Platform (SOSCIP) consortium. The calibration system was designed and developed on the SOSCIP's cloud analytic platform (SOSCIP 2019). The allocated cloud resource contains two VMs, or nodes, using the Linux Operating system, each with 24 computational cores, or CPUs. The calibration system has 196 GB of RAM and 2TB of a network's storage which is managed as centralized data storage to retrieve and store data by VMs using Network File System (NFS) in a faster and more efficient manner as compared to that of distributed data storages. The calibration process can be performed on a single or multiple VMs. In addition, the proposed cloud-based calibration system does not have a limitation for employing the maximum number of available computational cores if the computational resource on the cloud platform is extended.

### Optimization algorithms

#### Sequential uncertainty fitting (SUFI-2)

The SUFI-2 (Abbaspour *et al.* 2004, 2007) method was developed to address the degree of all uncertainties quantified by the *p*-stat measure which is the percentage of measured data grouped by the 95% prediction uncertainty (95PPU). The *r*-stat is another measure that quantifies the strength of the uncertainty analysis of a calibration from the average of the 95PPU band divided by the standard deviation of the measured data. The SUFI-2 method aims to detect the

majority of the measured data with the smallest uncertainty band. The 95PPU is calculated at the 2.5 and 97.5% levels of the cumulative distribution of an output variable obtained by using Latin Hypercube Sampling (LHS) (McKay *et al.* 1979). Therefore, it eliminates the lowest performing 5% of simulations. The value of the  $p$ -stat ranges between 0 and 100, and the value of  $r$ -stat ranges between 0 and infinity. A calibration that exactly corresponds to the measured data has a  $p$ -stat of 1 and  $r$ -stat of 0. LHS is a statistical method to sample evenly over the sample space from the random parameter values of a multidimensional distribution. An LH is considered as a predefined number of dimensions where each sample is the only one in each axis-aligned hyperplane that contains the sample. LHS is applied usually to reduce the computational time of running Monte Carlo simulations which can decrease the processing time by up to 50%.

### Dynamically dimensioned search

DDS (Tolson & Shoemaker 2007) is a heuristic optimization algorithm for calibration of watershed simulation models. The DDS optimization method was introduced for calibration problems that require a large set of decision variables whose lower and upper values are predefined. DDS aims to find an optimal solution within a user-defined number of simulations or, in general, function evaluations. Initially, the DDS algorithm explores globally the solution space and changes the search domain gradually to local searches when the number of function evaluations or simulations is reached to a predefined maximum number of iterations. In each iteration, the changes in the search domain are achieved dynamically and probabilistically by reducing dimensions (or the number of decision values) in search neighborhoods of a solution space. The decision values are the parameters of a watershed model which are adjusted. The probability that decision variable  $x$  is selected in iteration  $i$  is computed as follows:

$$P_x(i) = 1 - \frac{\ln(i)}{\ln(m)} \quad (1)$$

where  $m$  is the maximum number of function evaluations for a calibration. In Equation (1), the probability of adjusting

the value of a decision variable is reduced gradually by the increases in the number of iteration  $i$ .

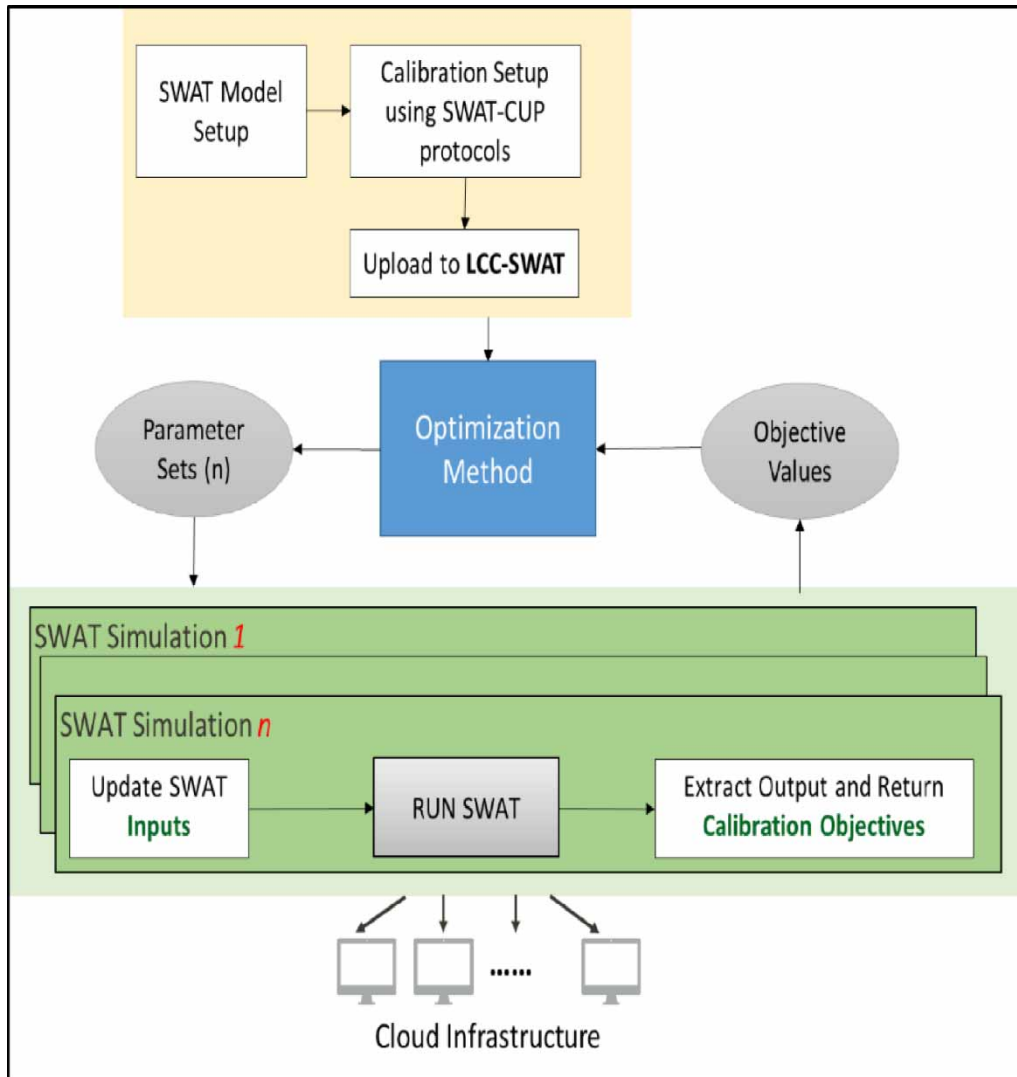
## THE LINUX-BASED CLOUD CALIBRATION SYSTEM FOR SWAT (LCC-SWAT) DEVELOPMENT

### System design

Figure 1 provides an overview of the LCC-SWAT calibration system workflow/design, runs in the Ubuntu 18.04 operating system. It has three core components. The first component is user-defined where a user creates a SWAT model, sets the calibration parameters using SWAT-CUP protocols and uploads the model and calibration setup files to the LCC-SWAT system on the cloud using Secure File Transfer Protocol (SFTP). The second component is the parallel optimization strategy. Two parallel optimization implementations are currently included in LCC-SWAT, which are discussed further in the ‘System implementation’ and ‘Parallelization of optimization algorithms’ sections. The final component of the system is the SWAT parallelization routines, which allow multiple SWAT simulations in the batch-parallel mode. If the optimization algorithm is set up to run  $n$  model runs in an iteration (i.e.,  $n$  parameter sets are provided by the optimization algorithm), these are equally distributed among available computing resources/cores.

### System implementation

The entire LCC-SWAT framework, including the two optimization methods (discussed in the ‘Parallelization of optimization algorithms’ section), their batch-parallel components and input/output communications with the SWAT model were implemented in C++. The actual model files are created and configured by the SWAT-CUP program under the Windows operating system on a personal computer. The entire model files are uploaded by the user to LCC-SWAT and copied to the internal network storage. VMs retrieve and store the original and updated model files from the network storage. In order to make the inputs and outputs of the parallel optimization methods compatible with SWAT-CUP, we



**Figure 1** | Overview of the Linux-based Cloud Calibration system for Soil and Water Assessment Tool (LCC-SWAT) system design.

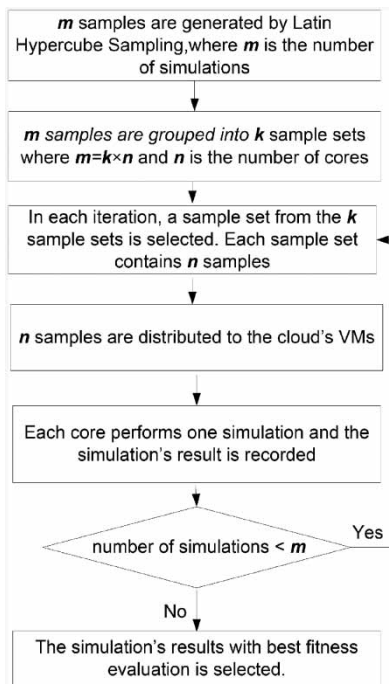
implemented (and replicated) the SWAT-CUP modules that edit model files and represent inputs and extract outputs to/from the SWAT program according to the desired time series of simulations, and other modules that describe the calibration workflow as defined in SWAT-CUP's manual (Abbaspour 2015). The Linux version of SWAT program and SWAT-CUP's 'swat\_edit' module were embedded and called as external routines at each step of simulation during the optimization. A number of SWAT-CUP's modules were not available for the Linux operating system. The functionality of these modules has to be replicated to work on Linux, e.g.,

'SUFU\_extract\_rch'. Hence, LCC-SWAT is compatible with SWAT-CUP and users of LCC-SWAT can formulate their SWAT model calibration using SWAT-CUP. The optimization methods proposed in LCC-SWAT do not employ a multi-objective function; however, multiple performance metric values (e.g.,  $R^2$ , NSE) are stored for each simulation during calibration for post-optimization analysis. LCC-SWAT can calibrate concurrently multiple watershed models by multiple users. However, the calibration times are increased differently based on how the cloud platform allocates the computational resources, the size of model files and the number of users.

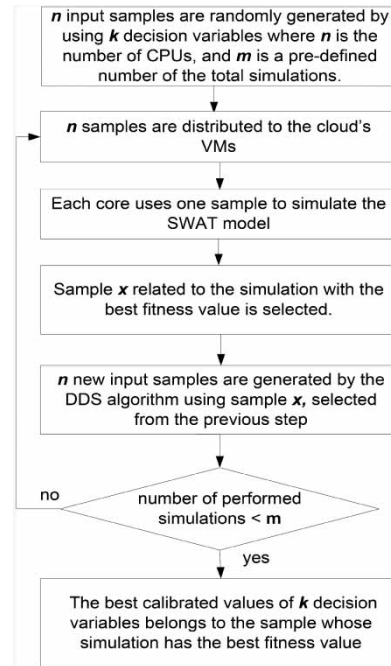
## Parallelization of optimization algorithms

The design of the cloud-based calibration system that employs the paralleled SUFI-2 algorithm is illustrated as follows. If  $n$  CPUs are employed in total to perform  $m$  simulations, the input decision values generated by LHS are grouped in  $k$  parameter sets where  $k$  is equal to  $m$  divided by  $n$ . In other words, the calibration process is performed in  $k$  iterations where  $n$  simulations are performed in each iteration. The results of simulations are collected and aggregated in each iteration. Next, after a predefined number of simulations is reached, the optimal calibrated values of the decision variables (or parameter set) are assigned to the simulation with the best fitness value as shown in Figure 2. Lastly, a comprehensive statistical report is computed for all simulations on the cloud platform for more robust statistical analyses.

The design of the cloud-based calibration system that uses iteratively the paralleled DDS optimization method is shown in Figure 3. Initially,  $n$  random samples or parameter sets are created with regards to the defined ranges of the SWAT model's decision variables, and constant  $n$  is also



**Figure 2** | An overview of the cloud calibration system using the Sequential Uncertainty Fitting (SUFI-2) optimization algorithm.



**Figure 3** | An overview of the cloud calibration system using the Dynamically Dimensioned Search (DDS) optimization algorithm.

equal to the number of computational cores. The samples are distributed to the cloud's VMs, and each core performs one simulation by using one of the parameter sets. The parameter set  $x$  whose simulation generates the best fitness value is identified. Then, the DDS optimization is performed  $n$  times on parameter set  $x$  to create  $n$  new parameter sets. The next iteration is started by repeating the distribution of the new parameter sets to the VMs. The calibration process is terminated when the maximum number of simulations is reached. Lastly, the parameter set  $x$  that is identified in the last iteration usually corresponds to the simulation with the best fitness evaluation in all iterations. By the paralleled DDS optimization, a calibration performed with a high number of simulations usually generates a more optimal parameter set compared to that of a calibration with a lower number of simulations.

## Testing the cloud calibration system for different SWAT models

In order to test the effectiveness of the developed cloud computing system (i.e., LCC-SWAT) as a function of increasing size and complexity of watersheds, three SWAT models

were set up, i.e., (i) a small agricultural Wigal Creek watershed (19 km<sup>2</sup>) (Zhang *et al.* 2020), (ii) a medium-sized Grand River Basin (6,542 km<sup>2</sup>) (Kaur *et al.* 2019) and (iii) a large-sized Canadian Great Lakes Basin (215,918 km<sup>2</sup>). High spatial resolution spatial dataset (digital elevation model, land-use and soil; Supplementary Table S1), daily meteorological dataset (precipitation and temperature; Supplementary Table S1) and crop-management data were sourced from different agencies and used during the process. The model setup resulted in 452, 2679 and 29449 HRUs, respectively, for Wigal, Grand River and Canadian Great Lakes SWAT models. Models were run for 1,000 simulations in monthly timescale for a period of 12 years in the cloud platform using the SUFI-2 optimization algorithm.

We tested LCC-SWAT on two criteria, i.e., (i) via analyzing calibration runtime of the framework with varying number of allocated processors (4–48) and nodes/VMs (1–2) and (ii) via analyzing comparative performance of the two algorithms currently implemented in the framework (DDS and SUFI-2).

The primary purpose of the calibration runtime comparison was to provide some insights and guidelines on identifying optimal computing resource allocation for LCC-SWAT. The proposed calibration system can be deployed on multiple VMs or nodes with a predefined number of processors. The number of cores allocated for any calibration job should be within the bounds specified in the MPICH2 package, an open-source implementation of the message passing interface (MPI) method for parallel programming in C++ (used in LCC-SWAT). Moreover, core allocation for a calibration job should also be within available computational resource limitations. For instance, we deployed and tested LCC-SWAT on the SOSCIP cloud platform, where two computing nodes were available with 24 cores each (48 cores in total). Hence, the maximum core allocation for calibration was 48 in our experiments. Ideally, a calibration process with  $m$  simulations (assuming all simulations can be executed simultaneously, i.e., in a single batch) and  $n$  core allocations where  $m$  is greater than  $n$  can be performed faster by specifying  $n$  (on a single or multi-node system) to be equal to the number of available physical cores (which were 48 for our experiments). However, parallel runtime performance, typically, does not scale proportionally with the number of allocated cores and can vary significantly. Variations in

runtime performance can be attributed to multiple reasons, e.g., physical cloud infrastructure and coding structure of the underlying simulation, etc. (Hadka & Reed 2015). Hence, we analyzed runtime performance of LCC-SWAT with a different number of core allocations to (i) deduce optimal number of core allocations for LCC-SWAT's deployment on SOSCIP and (ii) provide insights on deducing optimal core allocation for deployment on other cloud infrastructure. For this analysis, we tested the cloud-based calibration system using nodes in regular increments of four cores at both single and double node (VMs) configurations.

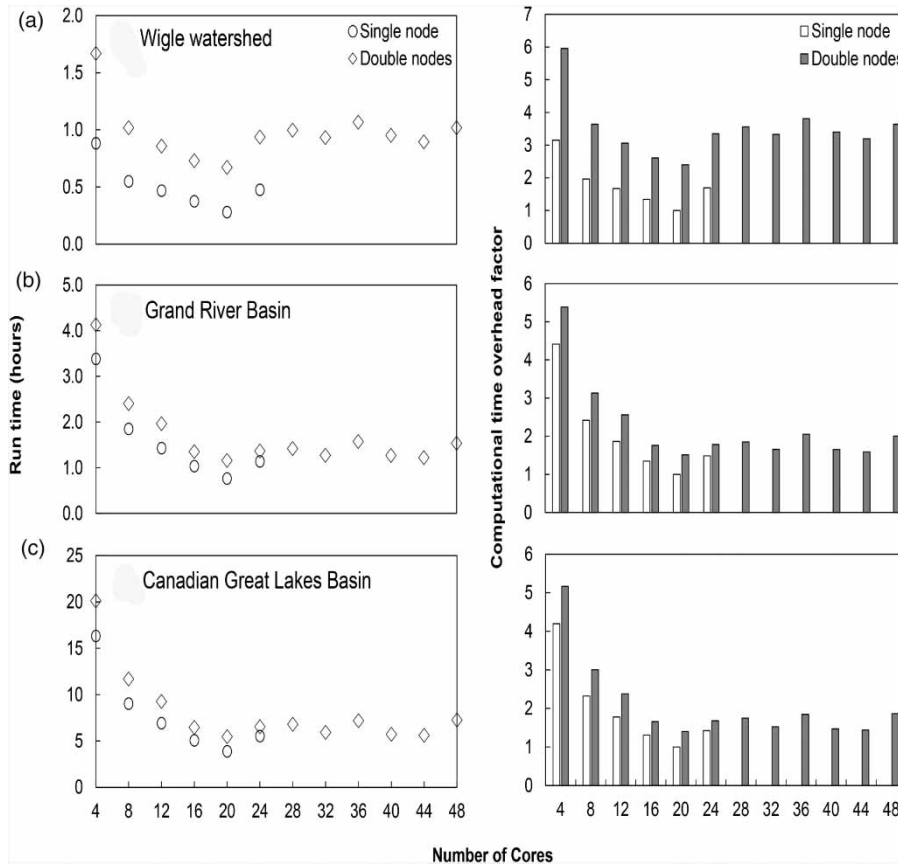
As different optimization algorithms have their own advantages in terms of converging to the global optima of multi-dimensional parameter search, it is important that more than one optimization algorithm is tested. Hence, the effectiveness of the above-mentioned optimization algorithms (SUFI-2 and DDS) was tested for the medium-sized Grand River Basin. Following a global sensitivity analysis using the SWAT-CUP, the 18 most sensitive SWAT parameters (Supplementary Table S2) were considered to optimize monthly streamflow measured at Grand River near Marsville, one of an upstream streamflow gauging station of the Grand River for an 8-year time period (2008–2015). The global sensitivity analysis regresses parameters generated using the LHS methodology (McKay *et al.* 1979) against a chosen objective function. We conducted the sensitivity analysis at monthly timescale using streamflow at the same Grand River near Marsville for the same 8-year time period (2008–2015) using the Nash–Sutcliffe Efficiency (NSE; Nash & Sutcliffe 1970) as the objective function. It should be noted that SWAT inherently runs in a daily timescale, as such, in our case, the daily simulations are aggregated in the monthly timescale. The range (maximum and minimum) of sensitive parameters values (Supplementary Table S2) were chosen based on similar reported works in cold-climate region basins (Faramarzi *et al.* 2015; Shrestha *et al.* 2017; Zhang *et al.* 2018; Kaur *et al.* 2019).

## RESULTS AND DISCUSSION

### Optimal number of cores

Figure 4 shows the evolution of model runtime when utilizing an increasing number of cores in single- and double-node





**Figure 4** | Runtime as the function of number of cores in single- and double-node configuration for the three Soil and Water Assessment Tool (SWAT) models. Also shown is the computation time overhead factor.

configuration for three separate SWAT models. As expected, the relative runtimes were higher for more complex SWAT models (e.g., Great Lakes Basin). Figure 4 indicates that utilization of a larger number of cores reduced, in our case, the LCC-SWAT calibration runtime. However, for both single and double node configurations, improvement (i.e., reduction) in calibration runtime continued until the utilization of 20 cores only. The runtime then increased when utilizing the maximum (24) cores on a single node. The same trend was also observed for calibration runs utilizing two nodes. Hence, we found 20 cores in both single and double node allocation scenarios to be the optimal core allocation.

As mentioned in the ‘Testing the cloud calibration system for different SWAT models’ section, runtime performance of parallel frameworks may not be proportional to the number of allocated cores, and this trend is also observed for LCC-SWAT. For instance, with single-node deployment, we observed that LCC-SWAT’s runtime

performance deteriorated when allocated processors increased from 20 to 24. One plausible reason for this deterioration is that when all 24 processors are allocated, the processors’ resources are shared for the execution of simulations and the operating system’s internal tasks and job scheduling. Thus, without idle processors, the concurrent execution of different tasks assigned to a node’s processors causes overloads and results in time latencies and increasing the overall runtime of the parallel calibration. The computational runtime improvement for a single node system may also be limited by hardware configuration. For instance, LCC-SWAT is deployed on a cloud infrastructure where the physical cores on a machine are doubled by hyper-threading technology to form logical cores which share execution, memory and I/O resources. Hence, given SWAT’s high I/O requirements (Zhang *et al.* 2017), runtime performance deterioration is expected for the cloud infrastructure employed in this study.

We also observed deterioration in runtime performance with the utilization of multiple nodes, i.e., the computational time of a calibration run on two nodes was more than that of the calibration on a single node with the same number of total processors. This deterioration was due to expected network latency and I/O operations required in communication between multi-node systems. Moreover, SWAT is an I/O intensive (Zhang *et al.* 2017) simulation model, and thus, parallel communication bottlenecks arising in parallel application of SWAT simulations in multi-node systems can easily supersede the potential advantage of the availability of additional cores.

Issues pertaining to linear scaling of runtime performance of SWAT parallelization frameworks, with the added number of cores, are also observed in other studies (Ercan *et al.* 2014; Zhang *et al.* 2015; Bacu *et al.* 2017). Furthermore, use of an increased number of cores has been debated from a cost-effectiveness point of view. For instance, Ercan *et al.* (2014) showed, with an experiment involving 256 cores, that use of 64 cores was the most desirable from the economical point of view.

Figure 4 also reports calculation time overhead, calculated as the ratio of the runtime in any configuration to the runtime for the optimal cores (20). This metric indicates that a cloud-based computing system (regardless of cloud infrastructure) could reduce calibration runtime of SWAT significantly (compared to desktop systems up to 8 cores). From the perspective of computational time overhead, the added value of such a cloud-based computing system is highlighted especially for the complex SWAT model, the Great Lakes Basin. For this SWAT model, the computation time overhead for all the configurations is lower than that observed for the less complex model (e.g., Wigle Watershed). For example, for two-node configuration and using 48 cores, the computation time overhead for the Wigle Watershed was 3.64 which reduced to 2.00 and 1.87 for the Grand River and Great Lakes Basin, respectively.

### Comparison of SUFI-2 and DDS algorithms

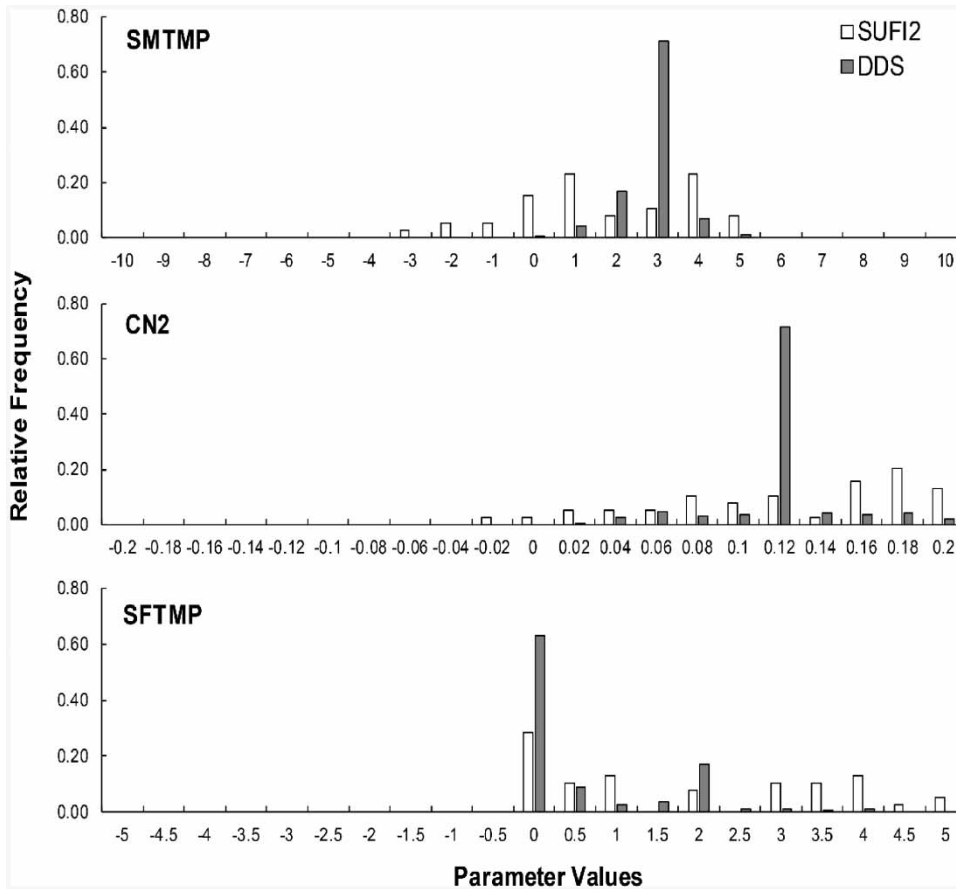
As a test of the cloud calibration system, both SUFI-2 and DDS algorithms were run for 1,000 simulations with NSE (Nash & Sutcliffe 1970) as the objective function to optimize monthly streamflow at Grand River near Marysville. While

the NSE was the objective function, we also calculated PBIAS and  $R^2$  to assign a qualitative rating to model simulation (Moriasi *et al.* 2015). Moreover, two performance aspects were considered during the comparative analysis of DDS and SUFI-2, i.e., (i) posterior parameter distributions (using behavioral solutions) obtained from both algorithms (see Section ‘Analyzing posterior parameter distributions’) and (ii) calibration statistics (of best calibrations found) and predictive uncertainty bounds obtained from both algorithms (see Section ‘Calibration statistics and predictive uncertainty’).

### Analyzing posterior parameter distributions

Figure 5 shows the posterior distributions of the three most sensitive parameters (identified during global sensitivity analysis; see Section ‘Testing the cloud calibration system for different SWAT models’), obtained from DDS and SUFI-2. Following Moriasi *et al.* (2015), all solutions with an NSE value more than 0.8 were considered as ‘behavioral solutions’, and the posterior distributions (represented by histograms in Figure 5) were estimated using behavioral solutions only. The results in Figure 5 show that DDS was successful in obtaining a narrower and more well-defined parameter distribution for all three parameters. The DDS-based posterior distribution of all parameters showed a clear and high relative frequency (~0.7), while the SUFI-2-based posterior distribution (Supplementary Table S2) seems to be spread in a wider range. In a stochastic modeling paradigm, the ability of an optimization algorithm to clearly identify an optimal range parameter is important given the issues related to parameter identifiability (Chavent 1985), especially for the SWAT model that is often regarded as an over-parameterized model (Nossent *et al.* 2011). Thus, the posterior distribution obtained from DDS is clearly better.

The relative superiority of DDS can be attributed to the algorithm’s iterative search dynamics, where, in each parallel simulation batch, new candidate calibration solutions are obtained by perturbing the best calibration found so far (see Figure 3 and Section ‘Dynamically dimensioned search’). Moreover, DDS only perturbs a subset of parameters in each algorithm iteration, and the number of parameters to be perturbed reduces as the algorithm progresses (see

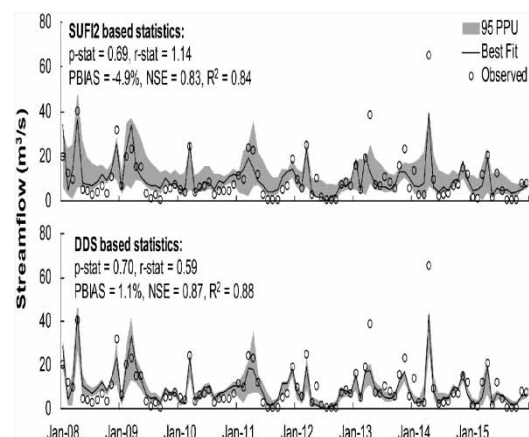


**Figure 5** | Posterior distribution of top three most sensitive parameters for all behavioral solutions from Sequential Uncertainty Fitting (SUFI-2) and Dynamically Dimensioned Search (DDS) optimization algorithms. SMTMP: Snow melt temperature, CN2: Curve number for moisture condition II, and SFTMP: Snow fall temperature.

Section ‘Dynamically dimensioned search’ and Equation (1)). This strategy is especially effective for calibration problems with many parameters (Asadzadeh & Tolson 2013). SUFI-2, on the other hand, uses LHS (McKay et al. 1979), which is a uniform stochastic space-filling design, and thus, posterior distributions obtained from SUFI-2 are more uniform. However, if the budget of functions evaluations for SUFI-2 is increased (from 1,000, which is not desirable for computationally expensive SWAT models), posterior distributions of parameters will become more well-defined and narrower.

### Calibration statistics and predictive uncertainty

Figure 6 shows the 95% predictive parameter uncertainty bands on the monthly streamflow for the 8-year period (2008–2015), for both DDS and SUFI-2. Owing to the



**Figure 6** | The 95% predictive parameter uncertainty (PPU) band on monthly streamflow of Grand River near Marsville. PBIAS: Percentage of bias, NSE: Nash-Sutcliffe Efficiency, and R2: Coefficient of determination.

better identifiability of parameters by the DDS algorithm over the SUFI-2 algorithm, the DDS 95% predictive parameter uncertainty band on monthly streamflow also

consistently out-performed the SUFI-2 algorithm (Figure 6). While the  $p$ -stat (percentage of the observations encapsulated in the 95% PPU band) values for both optimization algorithms are fairly comparable, there is a significant difference in the  $r$ -stat (the thickness of the 95% PPU band). The SUFI-2-based optimization resulted in a wider 95% PPU band ( $r$ -stat = 1.14), inferring higher uncertainty in simulated monthly streamflow at Grand River near Marysville. Furthermore, all the goodness-of-fit statistics (pertaining to the deterministic modeling paradigm) for the best-fit simulation also showed consistent underperformance of the SUFI-2 algorithm (compared to DDS). Following Moriasi *et al.* (2015) model performance criterion, the qualitative rating for the SUFI-2 based simulated monthly streamflow is 'good', while the same for the DDS-based simulation is 'very good'.

As stated in the 'Sequential uncertainty fitting (SUFI-2)' and 'Analyzing posterior parameter distributions' sections, SUFI-2 uses LHS (McKay *et al.* 1979), with uniform *a priori* distribution of model parameter from the defined range, to search for optimal solutions in a high-dimensional (i.e., with 18 parameters) parameter space. It is therefore evident that SUFI-2 optimization may not always reach the neighborhood of the global optima. Whereas the DDS algorithm initially explores globally in the solution space and changes the search domain gradually to local searches (by reducing the number of parameters to be perturbed; see Section 'Dynamically dimensioned search'). Hence, there is a higher chance that the DDS algorithm may find solutions close to the global optima. In our example case, the higher performance of the DDS algorithm may be related to the above-stated reasons. It should, however, be noted that further studies are needed to explicitly conclude the added advantage of one optimization algorithm over another. This cloud-based calibration (and uncertainty analysis) system indeed offers a platform to conduct such a computationally demanding task.

### Recommendations, limitations and future perspective of the work

It is well known that SWAT is a highly parameterized model (Nossent *et al.* 2011), and as such is highly I/O intensive. The LCC-SWAT system is flexible and can be deployed on any cloud platform (and with the different number of available

computational cores). However, increasing the number of cores may require the addition of more computational nodes or VMs. Since multiple nodes may be linked together by networks with I/O traffic affecting overall computational time, it is imperative that multi-node computational overhead is considered before deploying LCC-SWAT (and similar frameworks) on multi-node cloud infrastructure (as indicated in the results discussed in the 'Optimal number of cores' section). When frameworks similar to LCC-SWAT are deployed on multi-node cloud platforms, the architecture of the network's storage determines significantly the data storage, data retrieval and computational costs. Therefore, a very fast and dedicated network storage is a great advantage (and is highly recommended for LCC-SWAT deployment on multi-node systems) to boost the high demands of data access by VMs during parallel multi-node simulation runs in LCC-SWAT.

LCC-SWAT's implementation is modular and, thus, can incorporate extensions and modifications in future, to improve the system's ease-of-use and calibration performance. In this regard, three key extension/improvement avenues are (i) inclusion of more optimization algorithms, especially multi-objective algorithms for exploring calibration trade-offs, (ii) asynchronous parallel implementation of existing and new algorithms, for enhancing runtime efficiency and (iii) implementation of an interactive user interface (the user interface of LCC-SWAT is currently console-based). Given the computational time overhead induced by the cloud infrastructure and high I/O intensive nature of SWAT simulations, parallel efficiency of the LCC-SWAT framework may benefit significantly from the inclusion of asynchronous parallel optimization algorithms (Zhabitskaya & Zhabitsky 2012).

The user interface of LCC-SWAT can be enhanced from a console-based interaction to a more user-friendly visual interface, and the interface enhancement can be used as a blueprint when the LCC-SWAT system is planned to be distributed for client-based access through the internet. Development of a user-friendly interface may be especially important to enable broad uptake and use of the LCC-SWAT system. Many research and public organizations (e.g., U.S. Environmental Protection Agency (USEPA), Ontario Ministry of Natural Resources and Forestry (OMAFRA), etc.), use SWAT for watershed modeling (Francesconi *et al.* 2016) and watershed nutrient management. However, efficient and effective

SWAT calibration remains a challenge for such organizations. It is envisioned that LCC-SWAT could be used as a cloud calibration service tool by such organizations (especially organizations in Canada) in future.

---

## SUMMARY AND CONCLUSIONS

In this study, a cloud calibration and uncertainty analysis system offering two paralleled optimization algorithms (SUFI-2 and DDS) is developed for SWAT models, and deployed on the SOSCIP Cloud Analytic platform. The proposed cloud-based system, called LCC-SWAT, is a key contribution to the watershed calibration practice that allows parallel benchmarking of alternate parallel optimization algorithms on different cloud computing architectures. An illustration of the potential application of LCC-SWAT in parallel benchmarking of calibration algorithms is provided in this study via a comparison of parallel SUFI-2 and parallel DDS with a budget of 1,000 evaluations and with 20 cores each. Results show that the performance of DDS is better than SUFI-2.

Results of performance benchmarking of the LCC-SWAT system on both single (i.e., with one VM) and dual node (i.e., with two VMs) architectures are also provided with the application to three SWAT models of increasing complexities. Although a maximum of 48 cores in two VMs were available, results indicate that 20 cores in a single virtual machine is an optimal configuration (as per runtime perspective) for the cloud architecture tested in this study. However, for more complex watershed models, the runtime efficiency of multi-node systems improves since the computation time overhead reduces and core utilization improves. These results also indicated that an asynchronous parallel implementation may further improve efficiency and scalability of LCC-SWAT for multi-node systems. Moreover, the design of LCC-SWAT is modular and flexible; thus, other single- and multi-objective parallel algorithms can be added to enrich the system for efficiently solving future large-scale watershed model calibration problems. Finally, LCC-SWAT was also successfully integrated into the CANWET™ platform. The platform is designed to facilitate the use of modeling as a means of watershed management and policy testing. Further information is available at <https://www.gmnl.com/Greenland-Technologies-Group/CANWET.html>.

---

## ACKNOWLEDGEMENTS

The authors thank SOSCIP Smart Computing for Innovation for providing advanced computing platform.

---

## FUNDING

The corresponding author was supported with Natural Sciences and Engineering Research (NSERC) discovery grant (number: 2017-04400). Funding and access to parallel computing resources were provided to the project by SOSCIP – Smart Computing for Innovation. Greenland International Consulting Ltd was an in-kind contributor to the research.

---

## CONFLICTS OF INTEREST

The authors declare that there is no conflict of interest.

---

## AUTHOR CONTRIBUTION

P.D. and T.B. conceptualized the study. M.Z. developed the LCC-SWAT platform. N.K.S. tested the platform for various SWAT models for runtime, parameter identification, calibration and uncertainty analysis. M.Z., N.K.S. and T.A. drafted and revised the manuscript. P.D. and T.B. provided their comments and revision.

---

## DATA AVAILABILITY STATEMENT

Data cannot be made publicly available; readers should contact the corresponding author for details.

---

## REFERENCES

- Abbaspour, K. C. 2015 *SWAT Calibration and Uncertainty Programs – A User Manual*. Swiss Federal Institute of Aquatic Science and Technology (Eawag).
- Abbaspour, K. C., Johnson, C. A. & van Genuchten, M. T. 2004 *Estimating uncertain flow and transport parameters using a sequential uncertainty fitting procedure*. *Vadose Zone Journal* 3 (4), 1340–1352.

- Abbaspour, K. C., Yang, J., Maximov, I., Siber, R., Bogner, K., Mieleitner, J., Zobrist, J. & Srinivasan, R. 2007 **Modelling hydrology and water quality in the pre-alpine/alpine Thur watershed using SWAT**. *Journal of Hydrology* **333** (2–4), 413–430.
- Ahmadisharaf, E., Camacho René, A., Zhang Harry, X., Hantush Mohamed, M. & Mohamoud Yusuf, M. 2019 **Calibration and validation of watershed models and advances in uncertainty analysis in TMDL studies**. *Journal of Hydrologic Engineering* **24** (7), 03119001.
- Arnold, J. G., Srinivasan, R., Muttiah, R. S. & Williams, J. R. 1998 **Large area hydrologic modeling and assessment part I: model development**. *Journal of the American Water Resources Association* **34** (1), 73–89.
- Arnold, J. G., Kiniry, J. R., Srinivasan, R., Williams, J. R., Haney, E. B. & Neitsch, S. L. 2011 **Soil and Water Assessment Tool Input/Output File Documentation, Version 2009**. Agrilife Blackland Research Center, Temple, Texas.
- Arnold, J. G., Moriasi, D. N., Gassman, P. W., Abbaspour, K. C., White, M. J., Srinivasan, R., Santhi, C., Harmel, R. D., van Griensven, A., Van Liew, M. W., Kannan, N. & Jha, M. K. 2012 **SWAT: model use, calibration, and validation**. *American Society of Agricultural and Biological Engineers* **55** (4), 18.
- Asadzadeh, M. & Tolson, B. 2013 **Pareto archived dynamically dimensioned search with hypervolume-based selection for multi-objective optimization**. *Engineering Optimization* **45** (12), 1489–1509.
- Atsatryan, H., Narsisian, W. & Asmaryan, S. 2016 **SWAT hydrological model as a DaaS cloud service**. *Earth Science Informatics* **9** (3), 401–407.
- Bacu, V., Mihon, D., Rodila, D., Stefanut, T. & Gorgan, D. 2011 **Grid based architectural components for SWAT model calibration**. In *HPCS 2011 - International Conference on High Performance Computing and Simulation*, 4–8 July, Istanbul, Turkey, pp. 193–198.
- Bacu, V., Nandra, C., Stefanut, T. & Gorgan, D. 2017 **SWAT model calibration over Cloud infrastructures using the BigEarth platform**. In *2017 13th IEEE International Conference on Intelligent Computer Communication and Processing (ICCP)*, pp. 453–460.
- Beven, K. & Binley, A. 1992 **The future of distributed models: model calibration and uncertainty prediction**. *Hydrological Processes* **6** (3), 279–298.
- Borah, D. K., Ahmadisharaf, E., Padmanabhan, G., Imen, S. & Mohamoud, Y. M. 2019 **Watershed models for development and implementation of total maximum daily loads**. *Journal of Hydrologic Engineering* **24** (1), 03118001.
- Chappell, D. 2009 **Introducing Windows Azure**. David Chappell and Associates.
- Chavent, G. 1985 **On parameter identifiability**. *IFAC Proceedings Volumes* **18** (5), 531–536.
- Deb, K., Pratap, A., Agarwal, S. & Meyarivan, T. 2002 **A fast and elitist multiobjective genetic algorithm: NSGA-II**. *IEEE Transactions on Evolutionary Computation* **6** (2), 182–197.
- Devia, G. K., Ganasri, B. P. & Dwarakish, G. S. 2015 **A review on hydrological models**. *Aquatic Procedia* **4**, 1001–1007.
- enviroGRIDS 2009 **The enviroGRIDS Project**. Available from: <http://www.envirogrids.net/>
- Ercan, M. B. & Goodall, J. L. 2016 **Design and implementation of a general software library for using NSGA-II with SWAT for multi-objective model calibration**. *Environmental Modelling & Software* **84**, 112–120.
- Ercan, M. B., Goodall, J. L., Castronova, A. M., Humphrey, M. & Beekwilder, N. 2014 **Calibration of SWAT models using the cloud**. *Environmental Modelling & Software* **62**, 188–196.
- Famarzi, M., Srinivasan, R., Iravani, M., Bladon, K. D., Abbaspour, K. C., Zehnder, A. J. B. & Goss, G. G. 2015 **Setting up a hydrological model of Alberta: data discrimination analyses prior to calibration**. *Environmental Modelling & Software* **74**, 48–65.
- Francesconi, W., Srinivasan, R., Pérez-Miñana, E., Willcock, S. P. & Quintero, M. 2016 **Using the Soil and Water Assessment Tool (SWAT) to model ecosystem services: a systematic review**. *Journal of Hydrology* **535**, 625–636.
- Gorgan, D., Bacu, V., Mihon, D., Rodila, D., Abbaspour, K. & Rouholahnejad, E. 2012 **Grid based calibration of SWAT hydrological models**. *Natural Hazards and Earth System Sciences* **12** (7), 2411–2423.
- Gupta, H. V., Sorooshian, S. & Yapo, P. O. 1998 **Toward improved calibration of hydrologic models: multiple and noncommensurable measures of information**. *Water Resources Research* **34** (4), 751–763.
- Gupta, H. V., Kling, H., Yilmaz, K. K. & Martinez, G. F. 2009 **Decomposition of the mean squared error and NSE performance criteria: implications for improving hydrological modelling**. *Journal of Hydrology* **377** (1), 80–91.
- Gupta, H. V., Clark, M. P., Vrugt, J. A., Abramowitz, G. & Ye, M. 2012 **Towards a comprehensive assessment of model structural adequacy**. *Water Resources Research* **48** (8), W08301.
- Hadka, D. & Reed, P. 2013 **Borg: an auto-adaptive many-objective evolutionary computing framework**. *Evolutionary Computation* **21** (2), 231–259.
- Hadka, D. & Reed, P. 2015 **Large-scale parallelization of the Borg multiobjective evolutionary algorithm to enhance the management of complex environmental systems**. *Environmental Modelling & Software* **69**, 353–369.
- Han, F. & Zheng, Y. 2016 **Multiple-response Bayesian calibration of watershed water quality models with significant input and model structure errors**. *Advances in Water Resources* **88**, 109–123.
- Humphrey, M., Beekwilder, N., Goodall, J. L. & Ercan, M. B. 2012 **Calibration of watershed models using cloud computing**. In *2012 IEEE 8th International Conference on E-Science*, pp. 1–8.
- Joseph, J. F. & Guillaume, J. H. A. 2013 **Using a parallelized MCMC algorithm in R to identify appropriate likelihood functions for SWAT**. *Environmental Modelling & Software* **46**, 292–298.

- Kan, G., He, X., Li, J., Ding, L., Hong, Y., Zhang, H., Liang, K. & Zhang, M. 2019 Computer aided numerical methods for hydrological model calibration: an overview and recent development. *Archives of Computational Methods in Engineering* **26** (1), 35–59.
- Kaur, B., Shrestha, N. K., Daggupati, P., Rudra, R. P., Goel, P. K., Shukla, R. & Allataifeh, N. 2019 Water security assessment of the grand river watershed in Southwestern Ontario, Canada. *Sustainability* **11** (7), 1883.
- Leta, O. T., Nossent, J., Velez, C., Shrestha, N. K., van Griensven, A. & Bauwens, W. 2015 Assessment of the different sources of uncertainty in a SWAT model of the River Senne (Belgium). *Environmental Modelling and Software* **68**, 129–146.
- McKay, M. D., Beckman, R. J. & Conover, W. J. 1979 A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* **21** (2), 239–245.
- Moriyas, D. N., Gitau, M. W., Pai, N. & Daggupati, P. 2015 Hydrologic and water quality models: performance measures and evaluation criteria. *Transactions of the ASABE* **58** (6), 1763–1785.
- Nash, J. E. & Sutcliffe, J. V. 1970 River flow forecasting through conceptual models part I – a discussion of principles. *Journal of Hydrology* **10** (3), 282–290.
- Nossent, J., Elsen, P. & Bauwens, W. 2011 Sobol' sensitivity analysis of a complex environmental model. *Environmental Modelling & Software* **26** (12), 1515–1525.
- O'Donncha, F., Ragnoli, E., Venugopal, S., James, S. C. & Katrinis, K. 2016 On the efficiency of executing hydro-environmental models on cloud. *Procedia Engineering* **154**, 199–206.
- Rodila, D., Bacu, V. & Gorgan, D. 2012 Comparative parallel execution of SWAT hydrological model on multicore and grid architectures. *International Journal of Web and Grid Services* **8** (3), 304–320.
- Rodila, D., Ray, N. & Gorgan, D. 2015 Conceptual model for environmental science applications on parallel and distributed infrastructures. *Environmental Systems Research* **4** (1), 23.
- Rouholahnejad, E., Abbaspour, K. C., Vejdani, M., Srinivasan, R., Schulin, R. & Lehmann, A. 2012 A parallelization framework for calibration of hydrological models. *Environmental Modelling & Software* **31**, 28–36.
- Shirmohammadi, A., Chaubey, I., Harmel, R. D., Bosch, D. D., Muñoz-Carpena, R., Dharmasri, C., Sexton, A., Arabi, M., Wolfe, M. L., Frankenberger, J., Graff, C. & Sohrabi, T. M. 2006 Uncertainty in TMDL models. *Transactions of the ASABE* **49** (4), 1033–1049.
- Shrestha, N. K., Du, X. & Wang, J. 2017 Assessing climate change impacts on fresh water resources of the Athabasca River Basin, Canada. *Science of The Total Environment* **601–602**, 425–440.
- SOSCIP 2019 *Cloud Analytics*. Smart Computing for Innovation (SOSCIP). Available from: <https://www.soscip.org/2017impactreport/platforms/>.
- Tayfur, G. 2017 Modern optimization methods in water resources planning, engineering and management. *Water Resources Management* **31** (10), 3205–3233.
- Tolson, B. A. & Shoemaker, C. A. 2007 Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resources Research* **43** (1), W01413.
- van Griensven, A. & Meixner, T. 2007 A global and efficient multi-objective auto-calibration and uncertainty estimation method for water quality catchment models. *Journal of Hydroinformatics* **9** (4), 277–291.
- Vrugt, J. A. 2016 Markov chain Monte Carlo simulation using the DREAM software package: theory, concepts, and MATLAB implementation. *Environmental Modelling & Software* **75**, 273–316.
- Vrugt, J. A., Gupta, H. V., Bouten, W. & Sorooshian, S. 2003 A Shuffled Complex Evolution Metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. *Water Resources Research* **39** (8), 1201.
- Vrugt, J. A., ter Braak, C. J. F., Clark, M. P., Hyman, J. M. & Robinson, B. A. 2008 Treatment of input uncertainty in hydrologic modeling: doing hydrology backward with Markov Chain Monte Carlo simulation. *Water Resources Research* **44** (12), W00B09.
- Yang, J., Jakeman, A., Fang, G. & Chen, X. 2018 Uncertainty analysis of a semi-distributed hydrologic model based on a Gaussian Process emulator. *Environmental Modelling & Software* **101**, 289–300.
- Yapo, P. O., Gupta, H. V. & Sorooshian, S. 1998 Multi-objective global optimization for hydrologic models. *Journal of Hydrology* **204** (1), 83–97.
- Zhabitskaya, E. & Zhabitsky, M. 2012 Asynchronous differential evolution. In: *Mathematical Modeling and Computational Science. MMCP 2011. Lecture Notes in Computer Science* (G. Adam, J. Buša & M. Hnatič, eds). Springer, Berlin.
- Zhang, D., Chen, X. & Yao, H. 2015 Development of a prototype web-based decision support system for watershed management. *Water* **7** (2), 780–793.
- Zhang, D., Chen, X., Yao, H. & James, A. 2016 Moving SWAT model calibration and uncertainty analysis to an enterprise Hadoop-based cloud. *Environmental Modelling & Software* **84**, 140–148.
- Zhang, Y., Hou, J., Cao, Y., Gu, J. & Huang, C. 2017 OpenMP parallelization of a gridded SWAT (SWATG). *Computers & Geosciences* **109**, 228–237.
- Zhang, B., Shrestha, N., Daggupati, P., Rudra, R., Shukla, R., Kaur, B. & Hou, J. 2018 Quantifying the impacts of climate change on streamflow dynamics of two major rivers of the Northern Lake Erie Basin in Canada. *Sustainability* **10** (8), 2897.
- Zhang, B., Shrestha, N. K., Rudra, R., Shukla, R., Daggupati, P., Goel, P. K., Dickinson, W. T. & Allataifeh, N. 2020 Threshold storm approach for locating phosphorus problem areas: an application in three agricultural watersheds in the Canadian Lake Erie basin. *Journal of Great Lakes Research* **46** (1), 132–143.