

1 **Design and Implementation of a Software Library Integrating NSGA-II with**
2 **SWAT for Multi-Objective Model Calibration**

3

4 Mehmet B. Ercan, Research Assistant, Department of Civil and Environmental Engineering,
5 University of South Carolina, Columbia, SC, USA.

6

7 Jonathan L. Goodall, Associate Professor, Department of Civil and Environmental Engineering,
8 University of Virginia, Charlottesville, VA, USA.

9 Also

10 Jonathan L. Goodall, Adjunct Professor, Department of Civil and Environmental Engineering,
11 University of South Carolina, Columbia, SC, USA.

12

13 Corresponding Author: Jonathan L. Goodall, goodall@virginia.edu, PO Box 400742,
14 Charlottesville, Virginia 22904, (434) 243-5019

15

16

17

18 **Highlights**

19

20

21

22

- We present an open-source software library for calibration of SWAT models
- The library implements the NSGA-II multi-objective genetic algorithm
- The library is used to calibrate a SWAT model of the Upper Neuse Watershed, NC
- Results show how multi-objective optimization better constrains model calibration

23 **Abstract**

24 Calibrating watershed-scale hydrologic models remains a critical but challenging step in the
25 modeling process. The Soil and Water Assessment Tool (SWAT) is one example of a widely
26 used watershed-scale hydrologic model that requires calibration. The calibration algorithms
27 currently available to SWAT modelers through freely available and open source software,
28 however, are limited and do not include many multi-objective genetic algorithms (MOGAs). The
29 Non-Dominated Sorting Genetic Algorithm II (NSGA-II) has been shown to be an effective and
30 efficient MOGA calibration algorithm for a wide variety of applications including for SWAT
31 model calibration. Therefore, the objective of this study was to create an open source software
32 library for multi-objective calibration of SWAT models using NSGA-II. The design and
33 implementation of the library are presented, followed by a demonstration of the library through a
34 test case for the Upper Neuse Watershed in North Carolina, USA using six objective functions in
35 the model calibration.

36

37 Keywords: Multi-Objective Calibration; Genetic Algorithms; Watershed Modeling; SWAT;
38 NSGA-II

39

40 Software availability: The software is available free and open source on Github:

41 https://github.com/mehmetbercan/NSGA-II_Python_for_SWAT_model.

42 **1. Introduction**

43 The Soil and Water Assessment Tool (SWAT) is a widely used watershed model with
44 numerous applications around the world for water quantity and quality simulations (Cools et al.,
45 2011; Gassman et al., 2007; Liu et al., 2013). It can be classified as a semi-distributed
46 conceptual watershed model that is capable of running on a daily or sub-daily time step over long
47 time periods. SWAT is able to simulate large watersheds with different management scenarios
48 where the impact on water supply and non-point source pollution can be assessed (Arnold et al.,
49 1998). For SWAT and other similar watershed models, there are often hundreds of modeling
50 units in a model for a single watershed and dozens of model parameters used to describe
51 properties within the model. One of the modeler's most important and difficult tasks is to
52 calibrate these model parameters so that the model's output matches observational data such as
53 streamflow observations collected within the watershed.

54 Many algorithms and tools have been developed and applied for calibrating SWAT models.
55 SWAT-CUP represents one widely used tool in the SWAT community for applying calibration
56 algorithms to SWAT models. SWAT-CUP includes different calibration algorithms, as well as
57 routines for sensitivity analysis, validation, and uncertainty analysis of SWAT models
58 (Abbaspour et al., 2007). There are other procedures and algorithms developed in the scientific
59 community for calibration that have not yet been included in SWAT-CUP, but that would benefit
60 SWAT modelers. For example, SWAT-CUP does not include multi-objective calibration
61 approaches, nor does it include genetic algorithm calibration approaches (Abbaspour, 2013).
62 SWAT modelers, however, could benefit from these calibration procedures, especially for large
63 watersheds where multiple streamflow observations are available (Arnold et al., 1999; Bekele
64 and Nicklow, 2007; Kirsch et al., 2002; Santhi et al., 2001; White and Chaubey, 2005).

65 Genetic Algorithms (GAs) offer the ability to effectively solve highly non-linear
66 optimization problems and have been used for a variety of water resources challenges. Being an
67 evolutionary algorithm, GAs use principles of genetics and natural selection for optimization
68 (Haupt and Haupt, 2004). They are well suited for hydrologic models, which usually cannot be
69 adequately calibrated by gradient-based calibration algorithms. The objective function for each
70 solution in a GA can be evaluated in parallel computations, which provide computational
71 advantages (Zhang et al., 2013, 2012). The heuristic search procedure of GAs, relying on
72 stochastic search rules, increases the probability of finding non-unique solutions. Previous
73 studies have shown that these properties of GAs allow them to converge to optimal solutions for
74 a variety of problems (Winston et al., 2003) including the challenge of calibrating watershed-
75 scale hydrologic models (Arabi et al., 2006; Nicklow and Muleta, 2001).

76 Multi-objective calibration algorithms have been shown to increase model performance for
77 hydrologic models of large watersheds (Andersen et al., 2001). In contrast to the more widely
78 used single-objective calibration algorithms available to SWAT users now in tools like SWAT-
79 CUP, multiple-objective calibration better constrains the calibration process, resulting in a
80 calibrated model that better matches the physical conditions within the watershed (Niraula et al.,
81 2012). Watershed models may use multiple objective functions in a calibration procedure to
82 account for potentially competing objectives, even for cases when only a single streamflow
83 station is available for calibration (e.g., two objectives might be to match peak flows and
84 maintain annual water volume balance between the model and observations). They can also
85 allow modelers to take advantage of multiple observational time series (e.g., streamflow at two
86 or more locations in the watershed or streamflow and soil moisture observations at two or more
87 locations in the watershed).

88 There is a class a calibration routines that combine the benefits of both multi-objective and
89 genetic algorithm calibration approaches: the so called multi-objective genetic algorithms
90 (MOGAs). One of the most popular MOGAs is the Non-Dominated Sorting Genetic Algorithm
91 II (NSGA-II). NSGA-II is a fast and efficient population-based optimization technique that can
92 be parallelized. The algorithm has been shown to be superior to other MOGAs (Deb et al., 2002;
93 Zitzler et al., 2000) and it has the potential to reduce calibration time by efficiency in the
94 algorithm itself and its ability to easily be mapped to parallel computing resources (Deb et al.,
95 2002; Tang et al., 2006; Zitzler et al., 2000). The algorithm has significant improvements over
96 the original NSGA (Srinivas and Deb, 1994) including adding elitism, reducing the complexity
97 of the non-dominated sorting procedure, and replacing a sharing function with a crowded-
98 comparison function. The NSGA-II algorithm has also been shown to be an effective tool for
99 watershed model calibration (Bekele and Nicklow, 2007; Confesor and Whittaker, 2007; Hejazi
100 et al., 2008; Khu and Madsen, 2005; Shafii and Smedt, 2009; Zhang et al., 2010).

101 While NSGA-II has been used for calibrating watershed models, there is no known software
102 implementation of NSGA-II for calibrating SWAT models that is freely available to the
103 community. One study did report creating a multi-objective calibration tool for SWAT models
104 using NSGA-II (Bekele and Nicklow, 2007). However, based on personal communication with
105 the authors, the source code for this implementation is no longer available. The goal of this work,
106 therefore, is to create an open source and freely-available NSGA-II software library for SWAT
107 model calibration. We designed the tool to be library that can be used alone or incorporated into
108 other software tools. We specifically designed the software to be easily integrated into SWAT-
109 CUP given the popularity of this tool with the SWAT community. We chose to implement the

110 library using the Python programming language because of its growing popularity in the
111 scientific computing community.

112 In the remaining sections of this paper, we describe the algorithm for integrating NSGA-II
113 with SWAT for model calibration, then describe the design and implementation of the NSGA-
114 II/SWAT library, and finally present a test case application of the library for calibrating a SWAT
115 model of the Upper Neuse watershed in North Carolina. As part of this test case application, we
116 compare the results of the NSGA-II calibration to results from a single-objective calibration to
117 show the improvement obtained by using the multi-objective NSGA-II algorithm. We have
118 provided the source code for the NSGA-II/SWAT library as an open source and freely available
119 repository through GitHub: [https://github.com/mehmetbercan/NSGA-
120 II_Python_for_SWAT_model](https://github.com/mehmetbercan/NSGA-II_Python_for_SWAT_model).

121

122 **2. The NSGA-II Algorithm and its Integration with SWAT**

123 *2.1 Overall Process Flow*

124 In this section we explain the NSGA-II algorithm and how we integrated SWAT calibration
125 into the algorithm when designing the NSGA-II/SWAT library. For further detail on the NSGA-
126 II algorithm itself, readers are referred to (Deb et al., 2002). For convenience, we provide a
127 mapping between NSGA-II and SWAT calibration terminology in Table 1.

128

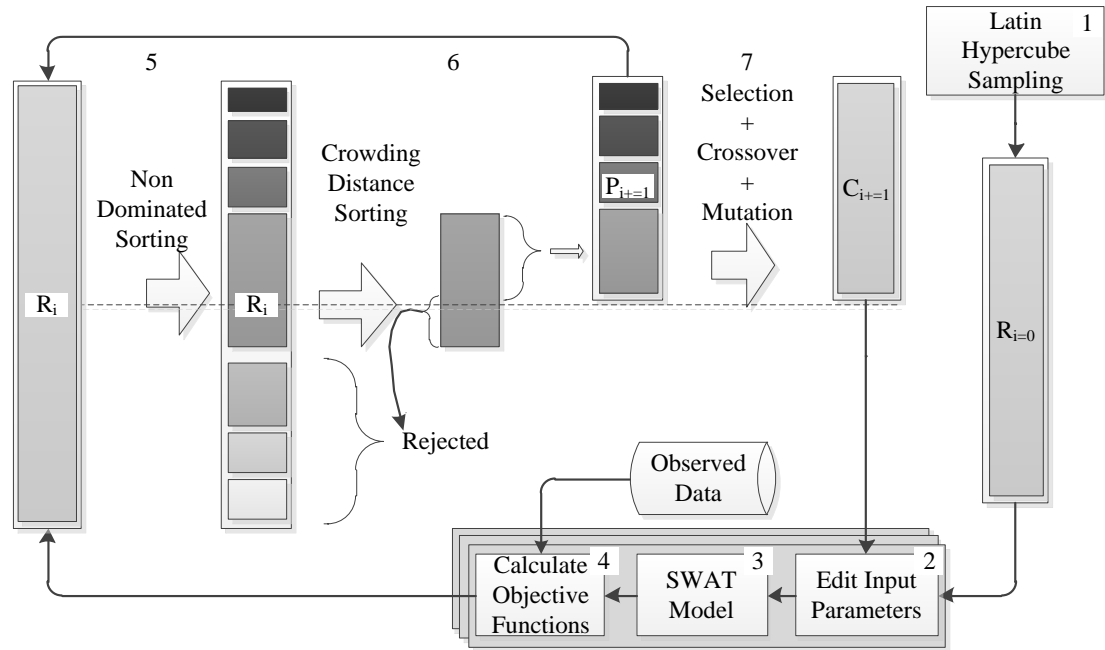
129 Table 1: Description of NSGA-II terms as they relate to SWAT calibration

NSGA-II Term	Description for Application to SWAT Calibration
Solution	An individual of a population that includes a SWAT calibration parameter set and NSGA-II processing data for the parameter set
Gene	The SWAT calibration parameter set that exists in a solution
Chromosome	An individual of a gene that represents a single SWAT calibration parameter
Binary Value	Binary representation of chromosome in a user defined number of bits

130

131 A standard NSGA-II process typically begins with a random parent population P_i (Deb et al.,
 132 2002). However, here we start with a Latin Hypercube Sampling (LHS) (See Step 1 in Figure 1)
 133 because better results have been achieved for SWAT models using this approach (Bekele and
 134 Nicklow, 2007). The LHS operator is executed first to create an initial combined population
 135 ($R_{i=0}$). We use the subscription “i” to represent a generation (iteration) number. The initial
 136 combined population must be at least twice as large the population size for reasons that will
 137 become clearer in forthcoming steps of the algorithm.

138



139

140 Figure 1: The NSGA-II algorithm for SWAT model calibration.

141

142 Each solution in the initial combined population ($R_{i=0}$) is considered to be a SWAT
 143 calibration parameter set. The SWAT input files are edited to include this solution, the model is
 144 executed, and the objective functions are evaluated using observational data and the SWAT
 145 model output data (See Steps 2-4 in Figure 1). These model runs can be performed in parallel for
 146 each solution within the population. Once this process has been completed, the solutions within
 147 the population (R_i) are ranked using the results of the objective function evaluation process and a
 148 non-dominating sorting approach (See Step 5 in Figure 1). Details of this non-dominating sorting
 149 approach are provided in Section 2.2.1.

150 The best performing solutions from R_i as determined by the non-dominating sorting
 151 approach are used to form the parent population (P_i). The number of solutions in the parent
 152 population is determined by the user defined population size. In the case of ties where multiple
 153 solutions exist with the exact same ranking at the cut-off point for creating P_i , a crowded distance

154 sorting operator is used to break the tie (See Step 6 in Figure 1). This operator is explained in
155 Section 2.2.2. In short, the solutions with the larger crowding distance value, which acts as a
156 dummy fitness in the sorting operator, are chosen to fill the remaining spots in P_i . Using the
157 parent population, a new child population (C_{i+1}) is determined through a selection, crossover
158 and mutation operator (See Step 7 in Figure 1), which is explained in Section 2.2.3. This entire
159 procedure is repeated until the termination criteria are met.

160

161 2.2 NSGA-II Operators

162 We provide in this section details for the specific operators used in the NSGA-II algorithm
163 that are mentioned in the previous section.

164 2.2.1 Non-Dominated Sorting

165 The non-dominated sorting operator is a process of ranking solutions that exist in the
166 combined population (R_i) (Deb et al., 2002; Srinivas and Deb, 1994). In this operator, the
167 objective functions are evaluated for given solutions to determine domination. Domination is
168 established when the objective function evaluations of a solution outperform all other solutions
169 with the same rank. The process terminates when all members of the combined population (R_i)
170 have been assigned a rank.

171 2.2.2 Crowding Distance Sorting

172 Crowding distance sorting is used to break ties for solutions with the same rank at the cut off
173 point for being included in the parent population (P_i) (Deb et al., 2002). First, the solutions in
174 that rank are sorted based on the value of an objective function. Then, a solution is selected and
175 the distance between that solution and each of the adjacent solutions is calculated. These

176 distances are normalized by dividing by the distance between the maximum and minimum value
177 of the objective function for all solutions. Finally, crowding distance for the solution is
178 calculated as the sum of the normalized distance for the adjacent solutions.

179 This process is repeated for all objective functions and the final crowding distance value for
180 a solution is the summation of crowding distances calculated for all objective functions. It is then
181 repeated for all solutions within the parent population. One exception is the maximum and
182 minimum solutions in a rank. Because they do not have adjacent solutions on both sides, they are
183 typically assigned an arbitrarily large distance value. When breaking ties, the preference is to
184 select solutions with a large crowding distance value, which means the solution has more distant
185 neighbors and selecting this solution helps to protect the diversity of the population.

186 2.2.3 Selection, Crossover, and Mutation

187 Selection is a process that chooses solutions from a parent population (P_{i+1}) that go into a
188 child population (C_{i+1}) based on non-dominated and crowding distance sorting values. It starts
189 by randomly selecting two solutions from P_{i+1} . Then, it selects the solution that has the smaller
190 rank. If two solutions have the same rank from non-dominated sorting, it selects the solution that
191 has the greater crowding distance value. This process continues until all spots in C_{i+1} are filled.

192 After completion of the selection process, the crossover process begins. There are two
193 techniques for the crossover operation: regular crossover and uniform crossover. In regular
194 crossover, each pair of adjacent solutions from C_{i+1} are progressively chosen. Then, a random
195 number is generated and compared to a crossover probability. If the random number is smaller
196 than the crossover probability, crossover occurs where chromosomes between the two solutions
197 flip for a randomly generated number of chromosomes.

198 Uniform crossover is different from regular crossover in that the crossover happens at a
199 binary level instead of at a solution level. The uniform crossover goes through all binary values
200 (0 or 1) (of chromosomes) for every evenly indexed $C_{i+=1}$ solution. Uniform crossover happens if
201 a random number is smaller than the crossover probability. In this case, the binary value is
202 replaced with the binary value from the corresponding next (oddly indexed) $C_{i+=1}$ solution.

203 Finally, mutation happens through $C_{i+=1}$ solutions at a binary level similar to uniform
204 crossover. The mutation process simply flips the binary value (from 1 to 0, or vice versa) if a
205 random number is smaller than the mutation probability.

206

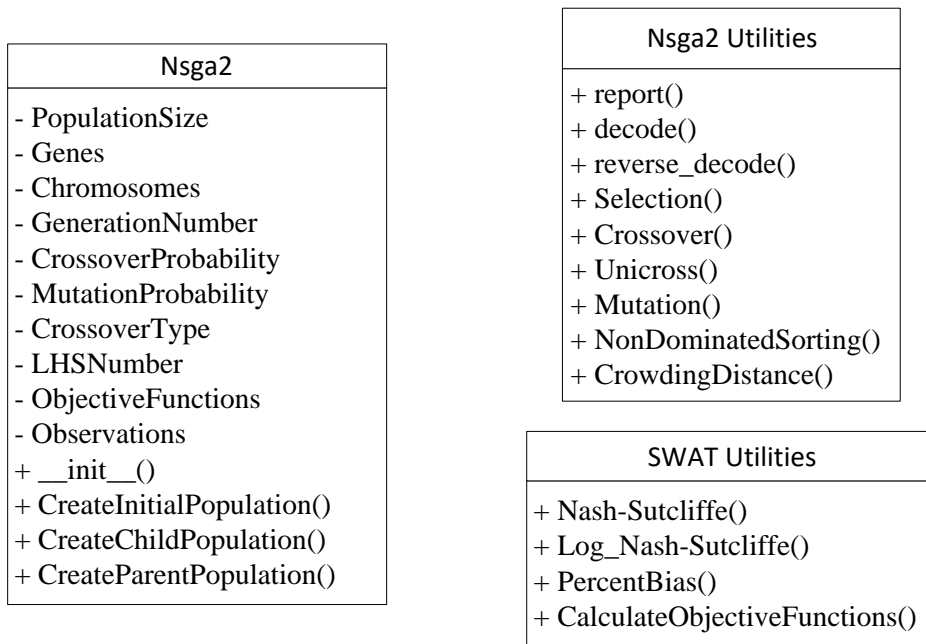
207 **3. Design and Implementation of the NSGA-II/SWAT Calibration Library**

208 The NSGA-II/SWAT calibration library implements the algorithm summarized in the prior
209 section where NSGA-II was integrated with SWAT for model calibration. The library was
210 designed as an object-oriented application programming interface (API) library and implemented
211 in the Python programming language because it is open source and widely used in scientific
212 communities. The library was tested against an established NSGA-II implementation written in
213 the C programming language (Deb et al., 2002) to ensure that it is able to reproduce the same
214 results. The library was designed to be compatible with SWAT-CUP (Abbaspour, 2013;
215 Abbaspour et al., 2007), which is a widely used tool for calibration of SWAT models, as
216 described later in this section.

217 *3.1 Class Diagram*

218 The NSGA-II/SWAT calibration library includes one main class called `nsga2` and two utility
219 classes for lower level NSGA-II and SWAT operations (Figure 2). The `nsga2` class is heart of
220 NSGA-II algorithm and includes operations such as creating child and parent populations.

221 During the initialization phase, the nsga2 class stores inputs such as population size, genes,
 222 chromosomes, and objective functions provided by the user. The nsga2 class offers two options
 223 for creating an initial combined population ($R_{i=0}$): (i) using the Latin Hypercube Sampling (LHS)
 224 method and (ii) reading the last generation from a previous calibration. The LHS method is
 225 included because, as stated earlier, it creates a better initial solutions for SWAT models (Bekele
 226 and Nicklow, 2007). On the other hand, reading the last generation from the previous calibration
 227 allows users to continue from previous but ultimately unsuccessful calibrations (for example, if a
 228 calibration fails to complete midway through the calibration process).



229
 230 Figure 2: The NSGA-II/SWAT calibration library design.

231
 232 The utility classes supplement the calibration process by providing lower-level functionality
 233 specific to the NSGA-II algorithm and for communication with SWAT. The nsga2 class uses
 234 nsga2 utilities to complete methods such as *Crossover()* or *Unicross()* required when creating
 235 child populations based on the user’s choice along with *Selection()* and *Mutation()* methods.

236 Similarly, creating a parent population requires methods like *NonDominatedSorting()* and
237 *CrowdingDistance()*, which are also implemented in the *nsga2* utility class. SWAT utilities are
238 used for objective function calculations using methods like *Nash-Sutcliffe()* and *PercentBias()*.
239 By separating the SWAT-specific functionality into its own class, our design goal was to provide
240 a pattern that could be repeated when expanding the library to support other hydrologic models.

241

242 *3.2 Application for SWAT Calibration*

243 To obtain SWAT model parameter values (genes), the binary values of chromosomes from
244 solutions of C_i go through a decoding process (*decode()*). Then, the SWAT model input files are
245 ready to be edited and executed to calculate objective functions using the SWAT utility class
246 method, *CalculateObjectiveFunctions()*. This method first creates a *model.in* file containing
247 genes. Then, it executes a batch file called *nsga2_mid.cmd* that creates the *model.out* file by
248 using the *model.in* file and the SWAT model engine. Finally, the *CalculateObjectiveFunctions()*
249 method uses the *model.out* file and calculates the objective function values by using other SWAT
250 utility functions such as *Nash-Sutcliffe()*. This process continues until each solution of C_i is
251 assigned objective function values.

252 The *nsga2_mid.cmd* file is a batch file that executes a series of commands for SWAT
253 calibration. It uses SWAT executable (*swat.exe*) and two Python scripts
254 (*SWAT_ParameterEdit.py* and *Extract_rch.py*) in order to create the *model.out* file. It first runs
255 *SWAT_ParameterEdit.py* to change SWAT model parameters based on information in *model.in*
256 file. Then, it executes *swat.exe* to execute the SWAT model using the parameter values included
257 in the *model.in* file. Finally, it runs *Extract_rch.py* to extract SWAT model outputs into
258 *model.out* file. The *nsga2_mid.cmd* file gives flexibility to edit the SWAT side of the calibration

259 procedure. To illustrate, inorganic nitrogen flux is the sum of nitrite (NO₂) and nitrate (NO₃),
260 which SWAT prints separately. Thus, an intermediate script could be inserted in *nsga2_mid.cmd*
261 file to sum these two nitrogen flux terms in *model.out* file for use in later calibration steps.

262

263 3.3 Compatibility with SWAT-CUP

264 The NSGA-II/SWAT calibration library was designed so that it can be integrated into
265 SWAT-CUP. First, we included a *Backup* folder as a reference to default parameter values as
266 done in SWAT-CUP. The input/output file and folder names were created following the SWAT-
267 CUP pattern. For example, the *SWATtxtInOut* folder contains the NSGA-II input and output
268 folders named *NSGA2.IN* and *NSGA2.OUT*. We further followed SWAT-CUP patterns by
269 creating files with the same structure. The calibration parameter definition file (*nsga2_par.def*) is
270 named with the calibration method and followed with *_par.def*. The structure of *nsga2_par.def*
271 file is defined as “*X_parameter.ext min max*” where the *X* defines the parameter editing method,
272 the *parameter* defines the SWAT parameter, the *ext* defines the extension of SWAT files, and the
273 *min* and *max* define the minimum and the maximum parameter limits.

274 In addition to the structure and naming conventions, internal parts of the NSGA-II/SWAT
275 library also follow the SWAT-CUP pattern. The *SWAT_ParameterEdit.py* script is equivalent to
276 *SWAT_edit.exe* of SWAT-CUP. Both scripts edit SWAT files based on the *model.in* file created
277 by the calibration algorithm. Also, the *Extract_rch.py* script is equivalent to SWAT-CUP’s
278 extracting script, *Extract_rch.exe*, which extracts SWAT outputs into *model.out* file in the
279 equivalent format. The batch file (*nsga2_mid.cmd*) mentioned in a prior section (which also
280 exists in SWAT-CUP) can be used to run extensive SWAT-CUP editing and extracting
281 executable files, rather than our parameter editing and extracting scripts. All these properties

282 were intentionally included to ease the integration of our software library into SWAT-CUP, a
283 future goal for this research.

284

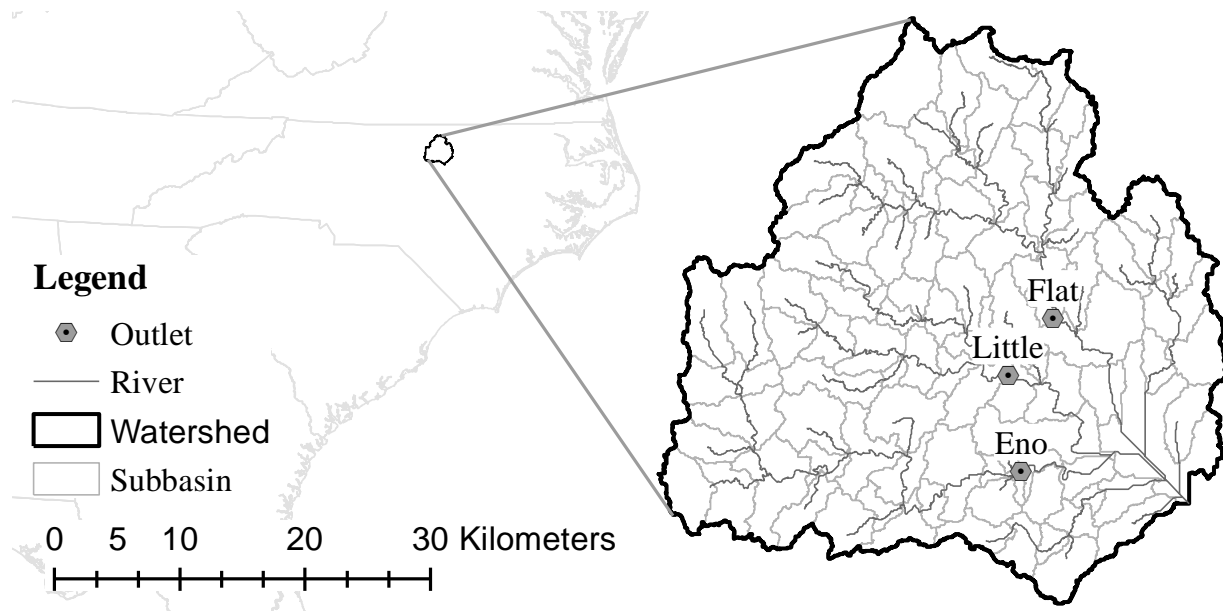
285 **4. Test Case**

286 The NSGA-II/SWAT library is demonstrated for a test case application using a SWAT
287 model of the Upper Neuse Watershed in North Carolina. The library is used to calibrate this
288 model to match streamflow records at three observation sites using two fitness criteria. In the
289 following subsections, we first briefly discuss how we created a SWAT model for Upper Neuse
290 watershed, second show how we used our NSGA II library to calibrate the SWAT model, and
291 third present the results of the calibration. The primary goal of this section is to illustrate how the
292 library would work for end users interested in applying the library to calibrate a SWAT model. A
293 secondary goal is to explore how the model calibration resulting from using the NSGA-II/SWAT
294 library compares to the widely used single-objective calibration strategy.

295 *4.1 Study Area and Model Preparation*

296 The Upper Neuse watershed (Figure 3) is a level-8 watershed that includes the Flat, Little,
297 and Eno River watersheds defined by the United States Geological Survey (USGS) codes
298 02085500, 0208521324 and 02085070, respectively. The study area has a mild climate and
299 gently rolling topography. The soil type of the watershed is dominated by silty clay and loam,
300 and the land cover of the watershed is dominated by forest and cultivated crops.

301



302

303 Figure 3: Study area: the Upper Neuse Watershed in North Carolina, USA.

304

305 Terrain and land cover data were obtained from the United States Geological Survey
 306 (USGS) National Elevation Dataset (NED) and the 2006 version of the National Land Cover
 307 Database (NLCD). Soil data were obtained from the State Soil Geographic (STATSGO) dataset
 308 provided by the United States Department of Agriculture (USDA). Air temperature, wind speed,
 309 and humidity were obtained from the National Climatic Data Center (NCDC). Precipitation data
 310 was obtained from National Weather Service (NWS) for Nexrad-derived rainfall estimates and
 311 from NCDC for gauge observed rainfall estimates. These two precipitation estimates were
 312 combined using the approach described by Ercan and Goodall (2012) to create a composite
 313 rainfall dataset for the watershed area. Lastly, daily average streamflow data from the USGS
 314 National Water Information System (NWIS) were downloaded using the Consortium of
 315 Universities for the Advancement of Hydrologic Science, Inc. (CUAHSI) Hydrologic
 316 Information System (HIS) (Tarboton et al., 2009).

317 We divided the watershed into subbasins based on the USGS streamflow station locations
318 and homogeneity of land characteristics. We used threshold values of 10% for soil, slope, and
319 land cover to reduce variability within the subbasins. The result was a total of 837 Hydrologic
320 Response Units (HRUs) for the 93 subbasins in the watershed, which is within the
321 HRU/subbasin ratio range recommended in SWAT documentation. The commonly used settings
322 were chosen to configure the model that include the Natural Resources Conservation Service
323 (NRCS) Curve Number (CN) surface runoff method, the Penman-Monteith potential
324 evapotranspiration method, and the variable storage channel routing method. The ArcSWAT
325 software program was used for much of the data preprocessing steps required to create the
326 model.

327 *4.2 Model Calibration*

328 Streamflow observations at the Flat, Little, and Eno watershed outlets were used in the
329 calibration. For each outlet, the Nash-Sutcliffe (E) and Percent Bias (PB) statistics were used as
330 measures of the goodness of fit. Therefore, the calibration used six objective functions (3 sites x
331 2 fitness). We ran Generalized Likelihood Uncertainty Estimation (GLUE) (Beven and Binley,
332 1992) available in SWAT-CUP to find the sensitivity of the flow parameters on streamflow
333 prediction. The six most sensitive parameters were chosen for model calibration with the
334 acceptable ranges and replacement operations shown in Table 2.

335

336 Table 2: Model parameters, their calibrated values, acceptable ranges, and replacement
 337 operations

Parameter	Value	Range	Operation
Alpha_Bf	0.99	0.01-1.00	Replaced
Cn2	0.07	±0.25	% Relative
Ch_K2	30.59	0.01-150.00	Replaced
Canmx	9.53	0.01-10.00	Replaced
Esco	0.94	0.01-1.00	Replaced
Sol_Aw c	-0.06	±0.25	% Relative

338

339 We used the following settings for calibrating the Upper Neuse watershed model with
 340 NSGA-II. The LHS size was set to 1000 and crossover probability was set to 0.5 using uniform
 341 crossover. The mutation probability and the seed for the random number generation were set to
 342 0.5. Population size and generation number were set to 80. Since our parameters do not have a
 343 wide range, we used 8 bits for binary crossover and mutations.

344 Figure 4 provides the pseudo code for the NSGA-II calibration to briefly illustrate how it
 345 was used in the case study. The first line initializes the `nsga2` class, which reads in the inputs
 346 from the `SWATtxtInOut` folder such as *PopulationSize*, *GenerationNumber* and *Observations*.
 347 Then the initial combined population is created followed by the generation loop. In the
 348 generation loop, the code first creates the parent population from the combined population.
 349 Second, it creates the child population using the parent population. Then the child population is
 350 used to run the SWAT model and the model's output is used to evaluate the objective functions.
 351 Finally, the parent and child populations are used to create the new combined population for the
 352 next generation. As seen in Figure 4, this library can easily be adapted to other watershed

353 simulation models by modifying the initialization method of the nsga2 class and the
354 *CalculateObjectiveFunctions()* process that exists in the SWAT utility class.

355

```
NSGAI = Nsga2.nsga2 (SWATtxtInOut)
Ri=0 = NSGAI.CreateInitialPopulation ()
Ri=0 = SWATUtilities.CalculateObjectiveFunctions (Ri=0)
FOR i = 0 to NSGAI.GenerationNumber
    Pi = NSGAI.CreateParentPopulation (Ri)
    Ci = NSGAI.CreateChildPopulation (Pi)
    Ci = SWATUtilities.CalculateObjectiveFunctions (Ci)
    Ri+1 = Pi + Ci
356 END FOR
```

357 Figure 4: The pseudo code for applying the NSGAI/SWAT library for calibrating the test case
358 SWAT model.

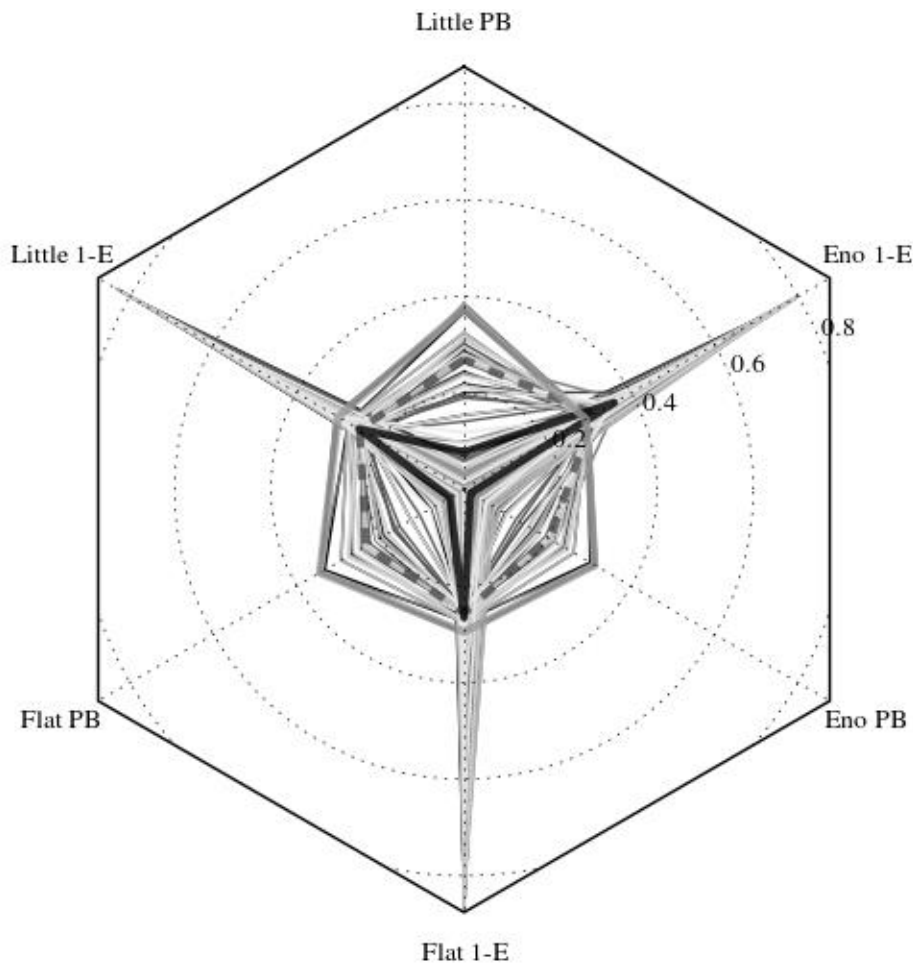
359

360 *4.3 Calibration Results*

361 The Pareto front solutions for the case study example are shown in Figure 5. There are six
362 objective functions for 80 solutions. The objective functions are percent bias (PB) and one minus
363 Nash-Sutcliffe (1-E) for the stations at the outlets of the Flat, Little and Eno watersheds. The
364 number of solutions is defined by the population size because all solutions in the final generation
365 are in the first front (ranking). A zero value on the figure indicates an optimal result while higher
366 values indicate worse model efficiency. The figure shows the range in performance of the three
367 watersheds in terms of PB and 1-E values. The values ranged between 0.00 and 0.39 for PB and
368 between 0.23 and 0.88 for E across the three observation sites.

369 We highlighted the tradeoffs in Figure 5. The thick black line shows the solution selected
370 with an equal weight for all objective functions, defining the best possible solutions considering
371 all three objective functions equally. When we put a large weight on the 1-E objectives, we get

372 the thick dashed grey line that slightly improves on 1-E values, but is worse for PB values. In the
373 last case with the thick grey line, we selected the lowest 1-E value (best E) for the Eno watershed
374 ignoring all other criteria. In this case, which represents calibration using a single objective
375 function, the E value improves for the Eno watershed as expected, but the other objective
376 functions, including PB for the Eno watershed, are worse compared to the equally weighted
377 multi-objective case.



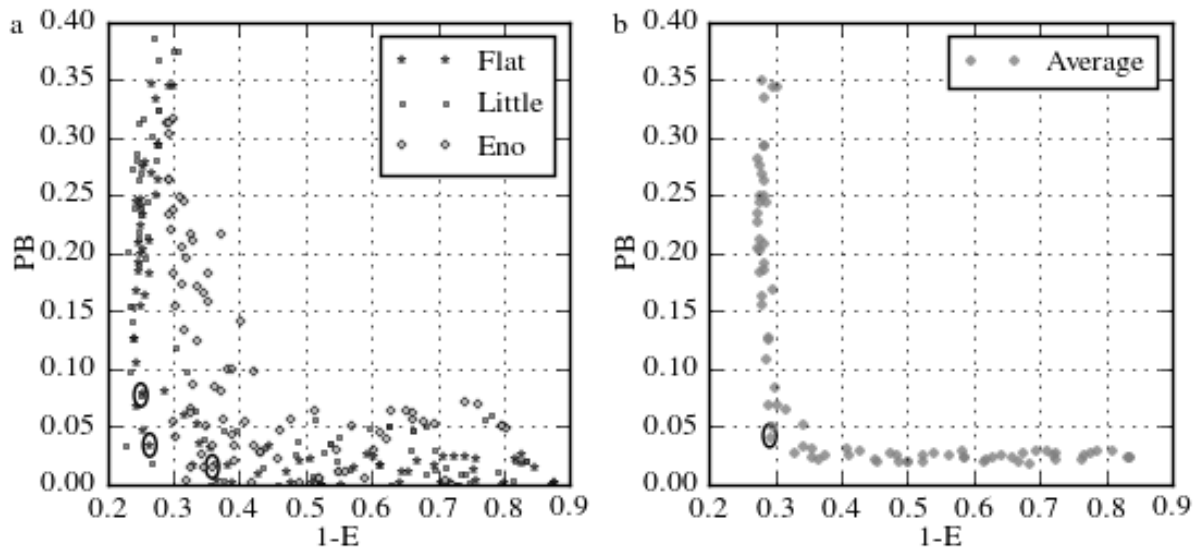
378
379 Figure 5: Six dimensional NSGA-II Pareto front.

380
381 For visualization of tradeoffs, we displayed the same Pareto front in Figure 5 using two
382 dimensional graphs. Because of difficulties of showing all six objective functions on a single

383 graph, we averaged fitness values over the Flat, Little and Eno watershed outlets in Figure 6b.
 384 Significant tradeoffs are illustrated between E and PB objective functions for the three outlets
 385 (Figure 6a) as was also shown by Bekele and Nicklow (2007). This illustrates the utility of a
 386 multi-objective calibration of SWAT models by attempting to balance multiple competing
 387 objectives when selecting optimal parameter sets.

388 The equally weighted objective functions are also highlighted in Figure 6. Better PB and 1-E
 389 values exist on Figure 6a. However, these values are connected to other objective functions that
 390 are much worse (e.g. the grey dashed and solid lines in Figure 5). Figure 6a indicates similar
 391 responses between the three watersheds, but a more significant relationship between the Flat and
 392 Little watersheds. This is expected as all the watersheds are in the same region and the Eno
 393 watershed is partially urbanized whereas the Flat and Little are not.

394



395

396 Figure 6: (a) NSGA-II Pareto front with (b) results averaged across the three watersheds.

397

398 Table 2 shows the parameter set values for the chosen solution (objective functions are
 399 equally weighted). We ran the SWAT model based on this solution and prepared the model

400 statistics against observations (Table 3). The daily and monthly statistics showed good
 401 agreement between simulated and observed streamflows for each site. PB values are considered
 402 to be “very good” for both the calibration and validation periods except for the Flat River
 403 watershed during the validation period, which is considered to be “good” (Moriassi et al., 2007).
 404 Monthly E values, on the other hand, were considered to be “good” for the calibration period and
 405 “very good” for the validation period (Moriassi et al., 2007). Lastly, daily statistics showed very
 406 good accuracy compared to previous SWAT studies (Gassman et al., 2007), indicating the
 407 strength of the calibration method.

408

409 Table 3: Results of the fitness values during the calibration and evaluation time periods for the
 410 Flat, Little, and Eno watersheds.

Watershed	2005-2008 ^a					2009-2012 ^b				
	E	E ^c	R ²	R ^{2c}	PB	E	E ^c	R ²	R ^{2c}	PB
Flat	0.74	0.73	0.75	0.74	0.04	0.62	0.8	0.62	0.82	-0.13
Little	0.75	0.72	0.76	0.73	0.08	0.61	0.8	0.61	0.81	-0.09
Eno	0.65	0.65	0.73	0.7	0.02	0.59	0.77	0.64	0.82	-0.11

a Calibration period

b Evaluation period

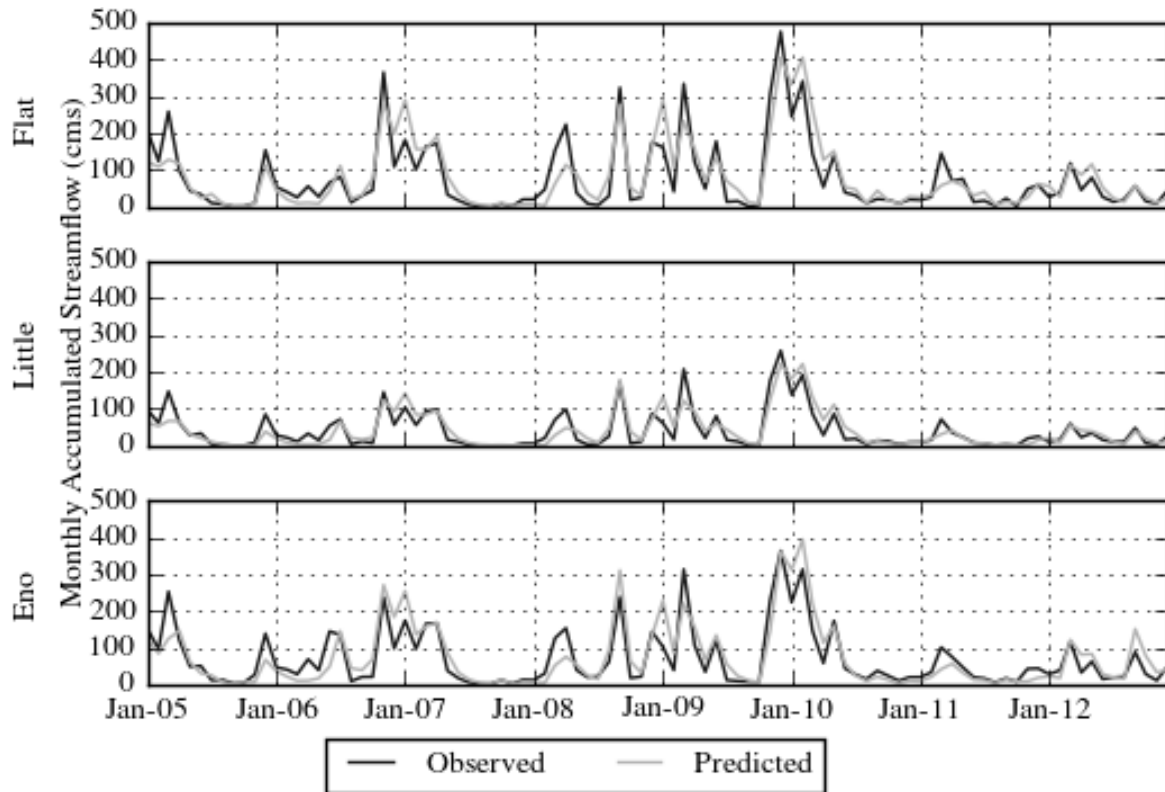
c Daily predicted and observed values aggregated to monthly

411

412 The solution with the equally weighted objective functions within the Pareto front is also
 413 illustrated in Figure 7. Similar to Table 3, the Little and Flat watersheds are slightly better at
 414 matching high flows (better E value) compared to the Eno watershed. All of the watersheds tend
 415 to underestimate streamflow for the calibration period and overestimate streamflow for the
 416 evaluation period. In general, the monthly accumulated streamflow values support the accuracy

417 of the model as both the calibration and evaluation periods generally fit well to observed
418 streamflow for all three sites.

419



420

421 Figure 7: Comparison of monthly simulated and observed streamflow.

422

423 Finally, we examined the solution with the best E value for Eno watershed (highlighted with
424 the thick grey line in Figure 5). This case is equivalent to single-objective calibration as we
425 selected a solution with regard to only one objective function and ignored all other objective
426 functions. When using this parameter set, the E value for the Eno watershed improved by 0.06
427 and 0.02 for calibration and validation periods, respectively, compared to the results when using
428 the parameter set from the equally weighted multi-objective solution. However, all other
429 statistics for the calibration and validation period for the three watersheds decreased when using

430 the parameter set from the single objective optimization. The magnitude of decrease in fitness
431 values was often similar to the gain in E for the Eno watershed. However, the PB values
432 deteriorated into an unacceptable model range (Moriassi et al., 2007) where PB values ranged
433 from 0.31 to 0.38 and 0.15 to 0.16 for calibration and validation periods, respectively, for the
434 three watersheds. This provides evidence to support the claim that multi-objective calibration
435 increases confidence in the model's predictive capabilities compared to using a single-objective
436 calibration routine.

437

438 **5. Conclusion**

439 The powerful Non-Dominated Sorting Genetic Algorithm II (NSGA-II) is a popular multi-
440 objective optimization genetic algorithm (MOGA) that has been shown to be effective for
441 calibrating watershed models including SWAT. Because there is no known software for linking
442 NSGA-II with SWAT for model calibration, we created an open source NSGA-II/SWAT library
443 using the Python programming language. We designed the library to be used either as a standard
444 alone tool for those experienced with Python, or as a library that can be incorporated by
445 developers into existing third-party Graphical User Interface (GUI) software tools. In particular,
446 a design goal was to allow for easy integration of the NSGA-II/SWAT library with the widely
447 used SWAT-CUP program that includes many algorithms for calibrating SWAT models, but
448 currently does not include the NSGA-II algorithm.

449 We demonstrated how the NSGA-II/SWAT library could be used through a test case
450 application for calibrating a SWAT model of the Upper Neuse Watershed in North Carolina. The
451 test case considered six objective functions: maximize Nash-Sutcliffe (E) and minimize Percent
452 Bias (PB) as the fitness coefficients for three streamflow stations located in the watershed. Six

453 model parameters were used in the calibration based on results obtained from using the GLUE
454 sensitivity analysis procedure. Results from applying the NSGA-II/SWAT library to this test
455 case showed large tradeoffs between fitness coefficients in the study watershed as illustrated in
456 the Pareto front. In general, the Eno watershed had lower E values compare to the other two
457 watersheds, and we suspect that this is due to urbanization within the Eno watershed that is not
458 present in the other two watersheds.

459 We chose the optimal parameter set from the Pareto front when weighting all objective
460 functions equally and used this parameter set to create the calibrated SWAT model. Results from
461 running the calibrated SWAT model during the time period used to calibrate the model were E
462 values ranging between 0.65 and 0.75 and PB values ranging between 0.02 and 0.08 for the three
463 streamflow stations used for calibration. The results from running the model during an
464 independent evaluation period not used for calibrating the model showed E values ranging
465 between 0.59 and 0.62 and PB values ranging between -0.13 and -0.09. All results for the
466 calibration and evaluation periods were considered to have satisfactory performance (Moriassi et
467 al., 2007) and improved results obtained from executing the SWAT model using an optimal
468 parameter set generated when considering only one of the six objective functions. Therefore, the
469 model calibration resulting from using the NSGA-II/SWAT library resulted in a well-calibrated
470 SWAT model that increases our confidence in the model's predictive capabilities compared to
471 the more common approach of using a single objective function.

472 The NSGA-II/SWAT tool was written to allow for easy expansion to include other
473 calibration algorithms and interfaces for other hydrological and environmental models that might
474 require multi-objective calibration. By having the source code in a public repository, the code
475 can be easily obtained and extended by others to include these enhancements. Furthermore, the

476 software was designed in a way so that it can be easily incorporated into front-end Graphical
477 User Interface (GUI) software tools, most notably SWAT-CUP. Future work incorporating the
478 library into SWAT-CUP in particular would be ideal so that it can leverage the existing data
479 visualization capabilities already available through SWAT-CUP and provide a new and powerful
480 calibration routine to SWAT-CUP users.

481

482 **Acknowledgments**

483 This work was funded in part by the US National Science Foundation under the award
484 CBET:0846244 and by the National Oceanic and Atmospheric Administration (NOAA) Global
485 Interoperability Program and the NOAA Environmental Software Infrastructure and
486 Interoperability Group.

487

488 **References**

- 489 Abbaspour, K.C., 2013. SWAT-CUP 2012: SWAT calibration and uncertainty programs-A user
490 manual, in: Swiss Federal Institute of Aquatic Science and Technology, Eawag.
- 491 Abbaspour, K.C., Vejdani, M., Haghghat, S., 2007. SWAT-CUP calibration and uncertainty
492 programs for SWAT, in: MODSIM 2007 International Congress on Modelling and
493 Simulation, Modelling and Simulation Society of Australia and New Zealand.
- 494 Andersen, J., Refsgaard, J.C., Jensen, K.H., 2001. Distributed hydrological modelling of the
495 Senegal River Basin---model construction and validation. *J. Hydrol.* 247, 200–214.
- 496 Arabi, M., Govindaraju, R.S., Hantush, M.M., 2006. Cost-effective allocation of watershed
497 management practices using a genetic algorithm. *Water Resour. Res.* 42.
- 498 Arnold, J.G., Srinivasan, R., Muttiah, R.S., Williams, J.R., 1998. Large area hydrologic
499 modeling and assessment part I: Model development1. *JAWRA J. Am. Water Resour.*
500 *Assoc.* 34, 73–89.
- 501 Arnold, J.G., Srinivasan, R., Ramanarayanan, T.S., DiLuzio, M., 1999. Water resources of the
502 Texas gulf basin. *Water Sci. Technol.* 39, 121–133.
- 503 Bekele, E.G., Nicklow, J.W., 2007. Multi-objective automatic calibration of SWAT using
504 NSGA-II. *J. Hydrol.* 341, 165–176.
- 505 Beven, K., Binley, A., 1992. The future of distributed models: model calibration and uncertainty
506 prediction. *Hydrol. Process.* 6, 279–298.

507 Confesor, R.B., Whittaker, G.W., 2007. Automatic Calibration of Hydrologic Models With
508 Multi-Objective Evolutionary Algorithm and Pareto Optimization. *JAWRA J. Am. Water*
509 *Resour. Assoc.* 43, 981–989.

510 Cools, J., Broekx, S., Vandenberghe, V., Sels, H., Meynaerts, E., Vercaemst, P., Seuntjens, P.,
511 Van Hulle, S., Wustenberghs, H., Bauwens, W., others, 2011. Coupling a hydrological
512 water quality model and an economic optimization model to set up a cost-effective emission
513 reduction scenario for nitrogen. *Environ. Model. Softw.* 26, 44–51.

514 Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic
515 algorithm: NSGA-II. *Evol. Comput. IEEE Trans.* 6, 182–197.

516 Ercan, M.B., Goodall, J.L., 2012. Estimating Watershed-Scale Precipitation by Combining
517 Gauge and Radar Derived Observations. *J. Hydrol. Eng.* 120807052807006.
518 doi:10.1061/(ASCE)HE.1943-5584.0000687

519 Gassman, P.W., Reyes, M.R., Green, C.H., Arnold, J.G., 2007. The soil and water assessment
520 tool: Historical development, applications, and future research directions. *Trans. ASABE*
521 50, 1211–1250.

522 Haupt, R.L., Haupt, S.E., 2004. *Practical genetic algorithms*. John Wiley & Sons.

523 Hejazi, M., Cai, X., Borah, D., 2008. Calibrating a watershed simulation model involving human
524 interference: an application of multi-objective genetic algorithms. *J. Hydroinformatics* 10,
525 97–111.

526 Khu, S.T., Madsen, H., 2005. Multiobjective calibration with Pareto preference ordering: An
527 application to rainfall-runoff model calibration. *Water Resour. Res.* 41.

528 Kirsch, K., Kirsch, A., Arnold, J.G., 2002. Predicting sediment and phosphorus loads in the Rock
529 River basin using SWAT. *Forest* 971, 10.

530 Liu, R., Zhang, P., Wang, X., Chen, Y., Shen, Z., 2013. Assessment of effects of best
531 management practices on agricultural non-point source pollution in Xiangxi River
532 watershed. *Agric. Water Manag.* 117, 9–18.

533 Moriasi, D.N., Arnold, J.G., Van Liew, M.W., Bingner, R.L., Harmel, R.D., Veith, T.L., 2007.
534 Model evaluation guidelines for systematic quantification of accuracy in watershed
535 simulations. *Trans. ASABE* 50, 885–900.

536 Nicklow, J.W., Muleta, M.K., 2001. Watershed management technique to control sediment yield
537 in agriculturally dominated areas. *Water Int.* 26, 435–443.

538 Niraula, R., Norman, L.M., Meixner, T., Callegary, J.B., 2012. Multi-gauge Calibration for
539 modeling the Semi-Arid Santa Cruz Watershed in Arizona-Mexico Border Area Using
540 SWAT. *Air, Soil Water Res.*

541 Santhi, C., Arnold, J.G., Williams, J.R., Hauck, L.M., Dugas, W.A., 2001. Application of a
542 watershed model to evaluate management effects on point and nonpoint source pollution.
543 *Trans. ASAE* 44, 1559–1570.

544 Shafii, M., Smedt, F. De, 2009. Multi-objective calibration of a distributed hydrological model
545 (WetSpa) using a genetic algorithm. *Hydrol. Earth Syst. Sci.* 13, 2137–2149.

546 Srinivas, N., Deb, K., 1994. Multiobjective optimization using nondominated sorting in genetic
547 algorithms. *Evol. Comput.* 2, 221–248.

548 Tang, Y., Reed, P., Wagener, T., others, 2006. How effective and efficient are multiobjective
549 evolutionary algorithms at hydrologic model calibration? *Hydrol. Earth Syst. Sci. Discuss.*
550 10, 289–307.

551 Tarboton, D.G., Horsburgh, J.S., Maidment, D.R., Whiteaker, T., Zaslavsky, I., Piasecki, M.,
552 Goodall, J., Valentine, D., Whitenack, T., 2009. Development of a community hydrologic
553 information system, in: 18th World IMACS Congress and MODSIM09 International
554 Congress on Modelling and Simulation, Ed. RS Anderssen, RD Braddock and LTH
555 Newham, Modelling and Simulation Society of Australia and New Zealand and
556 International Association for Mathematics and Comput. pp. 988–994.

557 White, K.L., Chaubey, I., 2005. Sensitivity analysis, calibration, and validations for a multisite
558 and multivariable swat model. *JAWRA J. Am. Water Resour. Assoc.* 41, 1077–1089.

559 Winston, W.L., Venkataramanan, M., Goldberg, J.B., 2003. Introduction to mathematical
560 programming. Thomson/Brooks/Cole.

561 Zhang, X., Beeson, P., Link, R., Manowitz, D., Izaurralde, R.C., Sadeghi, A., Thomson, A.M.,
562 Sahajpal, R., Srinivasan, R., Arnold, J.G., 2013. Efficient multi-objective calibration of a
563 computationally intensive hydrologic model with parallel computing software in Python.
564 *Environ. Model. Softw.* 46, 208–218. doi:10.1016/j.envsoft.2013.03.013

565 Zhang, X., Izaurralde, R.C., Zong, Z., Zhao, K., Thomson, A.M., 2012. EVALUATING THE
566 EFFICIENCY OF A MULTI-CORE AWARE MULTI-OBJECTIVE OPTIMIZATION
567 TOOL FOR CALIBRATING THE SWAT MODEL. *Trans. ASABE*, 55(5)1723-1731.

568 Zhang, X., Srinivasan, R., Liew, M. Van, 2010. On the use of multi-algorithm, genetically
569 adaptive multi-objective method for multi-site calibration of the SWAT model. *Hydrol.*
570 *Process.* 24, 955–969. doi:10.1002/hyp.7528

571 Zitzler, E., Deb, K., Thiele, L., 2000. Comparison of multiobjective evolutionary algorithms:
572 Empirical results. *Evol. Comput.* 8, 173–195.

573