ADVANCING HYDROLOGIC MODEL EVALUATION AND IDENTIFICATION USING MULTIOBJECTIVE CALIBRATION, SENSITIVITY ANALYSIS, AND PARALLEL COMPUTATION

A Thesis in
Civil Engineering
by
Yong Tang

© 2007 Yong Tang

Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

May 2007
The thesis of Yong Tang was reviewed and approved* by the following:

Patrick M. Reed  
Assistant Professor of Civil Engineering  
Thesis Advisor, Chair of Committee

Chris J. Duffy  
Professor of Civil Engineering

Thorsten Wagener  
Assistant Professor of Civil Engineering

Henry Lin  
Associate Professor of Hydropedology/Soil Hydrology

Peggy Johnson  
Professor of Civil Engineering  
Head of the Department of Civil and Environmental Engineering

*Signatures are on file in the Graduate School.
Abstract

This thesis work has comprehensively compared, developed, and implemented tools for advancing the evaluation and identification of hydrologic models including the lumped conceptual Sacramento Soil Moisture Accounting (SAC-SMA) model coupled with a snow accumulation and ablation model (SNOW-17), the distributed conceptual Research Distributed Hydrologic Model (HL-RDHM), and a semi-distributed version of the physical Penn State Integrated Hydrologic Model (PIHM). The model evaluation and identification tools addressed in this thesis include evolutionary multiobjective optimization algorithms and several sensitivity analysis methods implemented for distributed parallel computing systems. This thesis work was partitioned into four component studies. Study 1 assesses the efficiency, effectiveness, reliability, and ease-of-use of state-of-the-art evolutionary multiobjective optimization (EMO) tools when calibrating the SAC-SMA and the PIHM. This research proposes and demonstrates a formal metrics-based methodology for algorithm evaluation that clearly demonstrates their relative strengths and weaknesses. Understanding the relative strengths and weaknesses of the currently available EMO algorithms was important for Study 2 in which two parallelization schemes were developed to improve EMO algorithms’ performance in terms of their computational cost, their ability to identify high quality solutions and their robustness on a variety of applications including computer science test functions, hydrologic model calibration, and long-term groundwater monitoring design.

Beyond EMO algorithmic improvements, model evaluation and identification also requires a detailed understanding of hydrologic simulations’ sensitivities to guide model improvement, advance calibration strategies, and enhance our understanding of the key observations and processes controlling model behavior. Study 3 compares the repeatability, robustness, efficiency, and ease-of-implementation of four sensitivity analysis (SA) methods ranging from local analysis using parameter estimation software (PEST) to global approaches including regional sensitivity
analysis (RSA), analysis of variance (ANOVA), and Sobol’s method. The four SA tools were applied to the fully lumped SAC-SMA coupled with SNOW-17 using different model time steps and watershed locations. The results show that lumped model parameter sensitivities are heavily impacted by the choice of analysis method, model time interval, and local watershed characteristics. Study 4 extends Study 3 to advance distributed hydrologic model evaluation and identification using Sobol’s variance decomposition method since it was shown to be more robust and interpretable relative to the other sensitivity analysis methods tested.

Study 4 demonstrates a methodology that balances the computational constraints posed by global sensitivity analysis with the need to fully characterize the HL-RDHM’s sensitivities. The model’s sensitivities were assessed for long-term (annual and monthly) as well as short-term (events) forecasting periods. Overall, the results reveal that storage variations, spatial trends in forcing, cell-connectivity, and cell proximity to the gauged outlet are the four primary factors that control the HL-RDHM’s behavior. This study suggests that operational forecasts would benefit from the joint use of a robust sensitivity analysis framework directly integrated into new calibration methodologies. Overall, this thesis advances the analysis, formulation, and solution of hydrologic model evaluation and identification problems using multiple performance objectives and state-of-the-art algorithms implemented to exploit high-performance computing.
# Table of Contents

## List of Figures

x

## List of Tables

xvi

## Acknowledgments

xx

## Chapter 1 Introduction

1

## Chapter 2 Background

5

2.1 Multiobjective terminology and tools ........................................ 5

2.2 Multiobjective model calibration ............................................. 7

2.3 Parallel evolutionary multiobjective algorithms .......................... 9

   2.3.1 A brief introduction to parallel computing ........................... 9

   2.3.2 Parallel EMO algorithms ............................................. 11

      2.3.2.1 Master-Slave model ........................................ 11

      2.3.2.2 Multi-Population model ..................................... 12

2.4 Parameter sensitivity analysis ............................................. 14

## Chapter 3 Study 1: Assessing state-of-the-art multiobjective tools for hydrologic model calibration

16

3.1 Introduction ........................................................................... 17

3.2 Evolution-based multiobjective search ..................................... 18

   3.2.1 Epsilon Dominance NSGAII ($\varepsilon$-NSGAII) ...................... 18

   3.2.2 Strength Pareto Evolutionary Algorithm 2 (SPEA2) ............... 19

   3.2.3 Multiobjective Shuffled Complex Evolution Metropolis (MOSCEM-UA) ....................................................... 20

   3.2.4 Similarities and difference between the algorithms ............... 21

3.3 Case studies ........................................................................... 23
3.3.1 Case study 1: The test function suite
3.3.2 Case study 2: Leaf River watershed
3.3.3 Case study 3: Shale Hills watershed
  3.3.3.1 Integrated model description
  3.3.3.2 Problem formulation
3.4 Description of computational experiment
  3.4.1 Algorithm configurations and parameterizations
  3.4.2 Performance metrics
3.5 Results
  3.5.1 Optimization results for the test function suite
  3.5.2 Optimization results for the Leaf River case study
  3.5.3 Optimization results for the Shale Hills test case
3.6 Discussion
  3.6.1 Relative benefits and limitations of SPEA2
  3.6.2 Relative benefits and limitations of MOSCEM-UA
  3.6.3 Relative benefits and limitations of $\varepsilon$-NSGAII
3.7 Conclusions

Chapter 4 Study 2: Overcoming performance limits for multiobjective solution tools using parallelization strategies

4.1 Introduction
4.2 Methodology
  4.2.1 Evolutionary multi-objective optimization search
  4.2.2 The $\varepsilon$-NSGAII
  4.2.3 Parallelization strategies for the $\varepsilon$-NSGAII
    4.2.3.1 The Master-Slave $\varepsilon$-NSGAII
    4.2.3.2 The Multi-Population $\varepsilon$-NSGAII
  4.2.4 Case studies
    4.2.4.1 Case 1: Test problem DTLZ6
    4.2.4.2 Case 2: Model calibration in the Leaf River watershed
    4.2.4.3 Case 3: Long-term groundwater monitoring design
  4.2.5 Performance metrics
4.3 Results
  4.3.1 Optimization results for case study 1: DTLZ6
  4.3.2 Optimization results for case study 2: Leaf River calibration application
  4.3.3 Optimization results for case study 3: Long-term monitoring application
4.4 Discussion
Chapter 5  Study 3: Assessing sensitivity analysis methods to advance lumped watershed model identification and evaluation

5.1 Introduction ............................................................................................................. 88
5.2 Sensitivity analysis tools and sampling schemes .................................................. 89
   5.2.1 Overview ........................................................................................................... 89
   5.2.2 Sensitivity analysis tools .................................................................................. 91
      5.2.2.1 PEST ......................................................................................................... 91
      5.2.2.2 Regional sensitivity analysis using Latin hypercube sampling ................. 92
      5.2.2.3 Analysis of variance using iterated fractional factorial design sampling .... 93
      5.2.2.4 Sobol’s method using quasi-random sequence sampling ......................... 95
5.3 Overview of the lumped hydrologic models ......................................................... 97
   5.3.1 SNOW-17 ......................................................................................................... 97
   5.3.2 Sacramento soil moisture accounting model .................................................. 99
5.4 Case study ............................................................................................................... 100
   5.4.1 Juniata watershed description ......................................................................... 100
   5.4.2 Data set ........................................................................................................... 102
5.5 Computational experiment ..................................................................................... 103
   5.5.1 Model setup and parameterizations ................................................................. 103
   5.5.2 Objective functions ......................................................................................... 104
   5.5.3 Bootstrap confidence intervals ....................................................................... 104
   5.5.4 Evaluation of sensitivity analysis results ......................................................... 105
5.6 Results .................................................................................................................... 105
   5.6.1 Sensitivity results for PEST ............................................................................. 106
   5.6.2 RSA Results ................................................................................................... 108
   5.6.3 Sensitivity results for ANOVA ........................................................................ 111
   5.6.4 Sensitivity results for Sobol’s method ............................................................. 115
   5.6.5 Comparative summary of sensitivity methods ................................................ 118
5.7 Discussion ............................................................................................................... 122
5.8 Conclusions ............................................................................................................ 125

Chapter 6  Study 4: Advancing the identification and evaluation of distributed rainfall-Runoff models using Sobol’s global sensitivity analysis

6.1 Introduction ............................................................................................................ 128
6.2 Overview of the Hydrology Laboratory Research Distributed Hydrologic Model (HL-RDHM) ........................................ 130
6.2.1 Model structure ................................................. 130
6.2.2 The components of HL-RDHM ............................. 131
   6.2.2.1 SNOW-17 .............................................. 131
   6.2.2.2 Sacramento Soil Moisture Accounting (SAC-SMA) model ..................................................... 133
   6.2.2.3 Hillslope and channel routing model ................. 134
6.3 Sobol’s sensitivity analysis ........................................ 135
   6.3.1 Sobol’s method ............................................... 135
   6.3.2 Latin Hypercube Sampling (LHS) .......................... 137
6.4 Case study .......................................................... 138
   6.4.1 Juniata watershed description ............................ 138
   6.4.2 Data set ..................................................... 138
6.5 Computational experiment ......................................... 139
   6.5.1 Model setup ................................................ 139
   6.5.2 Test cases and parameterization .......................... 140
   6.5.3 Sensitivity analysis implementation ...................... 141
6.6 Results ............................................................. 143
   6.6.1 Annual sensitivities based on distributed forcing and lumped parameters ............................................. 143
   6.6.2 Monthly sensitivities based on distributed forcing and lumped parameters ........................................... 146
   6.6.3 Event sensitivities based on distributed forcing and distributed parameters ....................................... 149
   6.6.4 Verification of event analysis sensitivity rankings .... 153
6.7 Discussion and conclusions ......................................... 156

Chapter 7 Overview of thesis conclusions and overall contributions 160

Chapter 8 Implications and future work ............................. 165
   8.1 Implications .................................................... 165
   8.2 Future work .................................................... 167

Appendix A Source codes for sensitivity analysis tools ............ 171
   A.1 Public functions ............................................. 171
   A.2 Latin Hypercube Sampling ................................... 173
   A.3 Iterated Fractional Factorial Design ......................... 174
   A.4 Sobol’s random sequence .................................... 180
   A.5 Analysis of Variance .......................................... 181
List of Figures

2.1 Example illustration of the Pareto front for a convex, 2-objective minimization problem. The Pareto front is indicated by the bold curve. The full set of feasible solutions includes the Pareto front and the solutions within the non-bolded curve. .......................... 7

2.2 (a) Shared Memory Architecture; (b) Distributed Memory Architecture. ................................................................. 10

2.3 Master-Slave Model Paradigm. ......................................................... 12

2.4 Multi-Population Model Paradigm. ......................................................... 12

3.1 Domain decomposition of the Shale Hills test case ......................... 27

3.2 Illustration of the Shale Hills calibration period where a 100 hour warm up period was used. High flow and low flow classifications were made based on the points of inflection within the hydrograph. 29

3.3 Dynamic performance plot for the unary ε-indicator distance metric versus total design evaluations for the best performing configurations of the ε-NSGAII, SPEA2, and MOSCEM. Mean performance is indicated by a solid line, the standard deviation by a dashed line, and the range of performance by the shaded region. The plots were generated using 50 trials for each algorithm. .......................... 35

3.4 (a) Reference set generated for the Leaf River test case (b) the percentage of the reference set contributed by ε-NSGAII, SPEA2, and MOSCEM-UA. ......................................................... 37

3.5 Hydrograph ranges corresponding to reference set of the Leaf River test Case. The shaded area between dashed lines is the range of simulated hydrographs associated with the parameters groups in reference set. The bold line indicates the observed streamflow. ...... 38

3.6 Leaf River test case dynamic performance results for the unary ε-indicator distance metric versus total design evaluations. Mean performance is indicated by a solid line, the standard deviation by a dashed line, and the range of performance by the shaded region. The plots were generated using 50 trial runs for each algorithm. .. 39
3.7 Dynamic performance plots showing the best performing Leaf River trial runs for each algorithm. ........................................... 39

3.8 (a) Reference set for the Shale Hills test case (b) projections of the reference set onto the 2-objective planes to highlight the tradeoffs between the objectives. .................................................. 41

3.9 The percentages of the Shale Hills reference set contributed by ε-NSGAII, SPEA2, and MOSCEM-UA. ................................. 41

3.10 Hydrograph ranges corresponding to reference set of the Shale Hills test Case. The shaded area between dashed lines is the range of simulated hydrographs associated with the parameters groups in reference set. The bold line indicates the observed streamflow. ... 42

3.11 Shale Hills test case dynamic performance results for the unary ε-indicator distance metric versus total design evaluations. Mean performance is indicated by a solid line, the standard deviation by a dashed line, and the range of performance by the shaded region. The plots were generated using 15 trial runs for each algorithm. ... 43

3.12 Dynamic performance plots showing the best performing Shale Hills trial runs for each algorithm. ........................................... 43

4.1 Illustration of the ε-dominance concept. Step 1 (a), step 2 (b), and the final result (c) following the application of ε-dominance. Figure is adapted from [1]. ................................................................. 54

4.2 Master-Slave Model Paradigm .................................................. 57

4.3 Population-archive relationship for the MP parallelization scheme. 59

4.4 Pareto front of DTLZ6 and the non-dominated region of point A and B on the front. The shaded region is the non-dominated region of A and B. Adapted from [2]. ......................................................... 63

4.5 Reference set generated for the Leaf River test case where RMSE(T) are the errors for the Box-Cox transform of the hydrograph and RMSE(R) are the errors for the raw hydrograph. .................... 65

4.6 Dynamic success rate plots for (a) the MS and (b) the MP configurations of the ε-NSGAII. Success rates are computed as the percentage of 50 trials that were able to attain an ε-performance value of 0.9. The success rates are shown as a function of computing clock time and processor count. Please note that 50 trial runs were used at each processor count. ......................................................... 73
4.7 Dynamic performance plots of $\varepsilon$-indicator for (a) the MS and (b) the MP versions of the $\varepsilon$-NSGAII. Each random trial is indicated with a solid line and the shaded regions show the ranges in performance. The $\varepsilon$-indicator values are shown as a function of computing clock time and processor count. Please note that 50 trial runs were used at each processor count. .............................. 75

4.8 Dynamic success rate plots for (a) the MS and (b) the MP configurations of the $\varepsilon$-NSGAII. Success rates are computed as the percentage of 50 trials that were able to attain an $\varepsilon$-indicator value of 0.1. The success rates are shown as a function of computing clock time and processor count. Please note that 50 trial runs were used at each processor count. ........................................ 77

4.9 Joint plots of average speedup versus average solution quality for (a) the MS and (b) the MP configurations of the $\varepsilon$-NSGAII with respect to the $\varepsilon$-indicator metric. The averages are computed for 50 trial runs at each processor count. Speedups were computed as the ratio of the average serial time ($T_S$) versus the average parallel time ($T_P$) required to attain each level of $\varepsilon$-indicator. ...................... 78

4.10 Dynamic performance plots of $\varepsilon$-performance for the MS and MP versions of the $\varepsilon$-NSGAII. Mean performances are indicated as lines and the shaded regions show the ranges in performance. The $\varepsilon$-performance values are shown as a function of computing clock time and processor count. Please note that 50 trial runs were used at each processor count. ........................................ 80

4.11 Dynamic success rates for (a) the MS and (b) the MP configurations of the $\varepsilon$-NSGAII. Success rates are computed as the percentage of 50 trials that were able to attain an $\varepsilon$-performance value of 0.8. The success rates are shown as a function of computing clock time and processor count. ........................................ 81

4.12 Joint plots of average speedup versus average solution quality for (a) the MS and (b) the MP configurations of the $\varepsilon$-NSGAII with respect to the $\varepsilon$-performance metric. The averages are computed for 50 trial runs at each processor count. Speedups were computed as the ratio of the average serial time ($T_S$) versus the average parallel time ($T_P$) required to attain each level of $\varepsilon$-performance. ...................... 82
5.1 Major SNOW-17 processes and their corresponding parameters.
MBASE-Base temperature for snowmelt computations during nonrain periods (degc). NMF-Maximum negative melt factor (mme/degc/6hr). TIPM-Antecedent temperature index parameter. PLWHC-Percent (decimal) liquid-water holding capacity. DAYGM - Constant amount of melt which occurs at the snow-soil interface whenever snow is present (mm). The full description of other parameter names can be found in section 5.3.1 and in Table 5.2. Shaded boxes represent the states or processes.

5.2 Major SAC-SMA processes and their corresponding parameters.
RIVA-Riparian vegetation area. SIDE-Ratio of deep recharge to channel base flow. RSERV-Fraction lower zone free water not transferable to tension water. The full description of other parameter names can be found in the table 5.2. The parameters in the shaded boxes pertain to storages or states.

5.3 Sub-watersheds in the Juniata river basin

5.4 Hydrologic conditions of headwater sub-watersheds in the Juniata River basin.

5.5 RSA (Regional Sensitivity Analysis) plot for Snow17 parameters in the SPKP1 watershed. The objective function is RMSE based on a 1-hour time interval.

5.6 RSA (Regional Sensitivity Analysis) plot for SAC-SMA parameters in the SPKP1 watershed. The objective function is RMSE based on a 1-hour time interval.

5.7 (a) ANOVA second order parameter interactions based on the RMSE measure. (b) ANOVA second order parameter interactions based on the TRMSE measure. Circles represent statistically significant F-values defined using the threshold value of 3.32. The color legends and shading represent the F-value magnitudes and ranges.

5.8 (a) Second order parameter interactions based on the RMSE measure computed using Sobol’s method. (b) Second order parameter interactions based on the TRMSE measure computed using Sobol’s method. Circles represent interactions that contribute at least 1% of the overall model output variance. The color legends and shading represent the Sobol indices’ magnitudes and ranges.

5.9 (a) Comparative summary of sensitivity classifications based on the high-flow RMSE model performance objective. (b) Comparative summary of sensitivity classifications based on the high-flow TRMSE model performance objective.
5.10 Example illustration of an independent test of the sensitivity classifications found for the SPKP1 watershed’s model at the 1-hour timescale. The scatter plots show the RMSE of streamflow predictions. Set 1 consists of 1000 randomly drawn Latin hypercube samples. Set 2 is composed of constant reference values for sensitive parameters and random samples of the remaining insensitive parameters. In Set 3 sensitive parameters are allowed to vary randomly and insensitive parameters are set to constant reference values. Term $r$ represents correlation coefficient.

6.1 HL-RDHM model grid for the Saxton (SXTP1) and Spruce Creek (SPKP1) watersheds.

6.2 Hydrographs for the two studied watersheds from year 2001 to year 2003. (a) Hydrograph for SPKP1. (b) Hydrograph for SXTP1.

6.3 Hydrographs for the two analyzed events in SPKP1. (a) Hydrograph for May 2002 event. (b) Hydrograph for September 2003 event.

6.4 Monthly HL-RDHM sensitivities using (a) Sobol’s first order indices and (b) Sobol’s total order indices. Sobol’s indices were computed using the RMSE measure and an hourly model time step. Triangles represent highly sensitive parameters that contribute at least 10% of the overall model output variance. Circles represent sensitive parameters that contribute at least 1% of the overall model output variance. The color legends and shading represent the Sobol indices’ magnitudes and ranges. Each row represents one month and each column represents one parameter. January and February of 2001 are missing because they represent the model warm up period.

6.5 Spatial distribution of the total event precipitation and cell-level sensitivities for the SPKP1 watershed. The May 2002 event is represented by (a) its spatial precipitation distribution, (b) the first order Sobol’s indices for each model cell, and (c) the cell level interactions. The September 2003 event is represented by (d) its spatial precipitation distribution, (e) the first order Sobol’s indices for each model cell, and (f) the cell level interactions. Note cell level interactions were computed as the difference between each cell’s total order and the first order Sobol’s indices. The cell-level Sobol’s indices were computed by summing over all of individual parameter indices analyzed in each cell. The arrows in the cells designate surface flow directions.
6.6 Spatial distribution of the total order sensitivity indices of the top three most sensitive parameters on each model grid cell. The spatial distributions of the May 2002 event sensitivities are plotted for the (a) the model’s upper zone tension water storage (UZTWM), (b) the model’s fraction of percolating water (PFREE), and (c) the model’s lower zone tension water storage (LZTWM). The spatial distributions of the September 2003 event sensitivities are plotted for the (d) the model’s upper zone tension water storage (UZTWM), (e) the model’s fraction of percolating water (PFREE), and (f) the model’s lower zone tension water storage (LZTWM). The arrows in the cells designate surface flow directions.

6.7 Verification plots for event analysis sensitivity rankings based on hourly model time steps for the HL-RDHM. The scatter plots show the RMSE of streamflow predictions. Set 1 consists of 1000 randomly drawn Latin hypercube samples for the 13 SAC-SMA parameters analyzed for the SPKP1 watershed model. Set 2 consists of random samples for the subset of model parameters composed by top 6 most sensitive SAC-SMA parameters perturbed across all model cells. Set 3 consists of random samples for all 13 SAC-SMA parameters analyzed using only the top 15 most sensitive model cells. Set 4 consists of random samples for the subset of model parameters representing the top 6 most sensitive parameters on the top 15 most sensitive model cells. Term $r$ represents correlation coefficient.
List of Tables

3.1 Suite of Test Functions ................................................. 24
3.2 Parameters being optimized in the Shale Hills case study ............ 28
3.3 Test function results for the ratios of top trial runs for each con-
figuration of the algorithms based on the binary $\varepsilon$-indicator metric
ranking. The values highlighted by bold font are the best values
among the configurations within a specific algorithm, the values
indicated by bold font with underscore are the best values across
algorithms. ................................................................. 33
3.4 Averages and standard deviations of the unary metrics for each al-
gorithm’s best configuration. AVG stands for mean, STD stands for stan-
dard deviation, and bolded entries highlight the best value
attained. ................................................................. 34
3.5 Leaf River case study’s ratios of top trial runs for each configura-
tion of the algorithms based on the binary $\varepsilon$-indicator metric ranking.
The best performing algorithm is highlighted in bold. .................. 37
3.6 Leaf River case study’s results for the averages and standard devi-
ations of the unary metrics for each algorithm configuration. AVG
stands for mean, STD stands for standard deviation, and bolded
entries highlight the best value attained. ............................ 37
3.7 Shale Hills case study’s ratios of top trial runs for each configura-
tion of the algorithms based on the binary $\varepsilon$-indicator metric ranking.
The best performing algorithm is highlighted in bold. .................. 42
3.8 Shale Hills case study’s results for the averages and standard devi-
ations of the unary metrics for each algorithm configuration. AVG
stands for mean, STD stands for standard deviation, and bolded
entries highlight the best value attained. ............................ 42
4.1 Summary of parameter settings both MS and MP versions of the
algorithm and metric calculations. NFE = Number of Function
Evaluations; P = Number of Processors. .............................. 71
4.2 DTLZ6 case study averages (AVG) and standard deviations (STD) in terms of the convergence (Conv.), diversity (Div.), \(\varepsilon\)-indicator (Eind.), and \(\varepsilon\)-performance (Eperf.) metrics. All metrics were computed using 50 random trials. MS and MP designate master-slave and multiple-population versions of the \(\varepsilon\)-NSGAII. The results are shown for 1, 2, 4, 8, and 16 processors. The best overall metrics are shown in underlined bold. 

4.3 Leaf River case study averages (AVG) and standard deviations (STD) in terms of the convergence (Conv.), diversity (Div.), \(\varepsilon\)-indicator (Eind.), and \(\varepsilon\)-performance (Eperf.) metrics. All metrics were computed using 50 random trials. MS and MP designate master-slave and multiple-population versions of the \(\varepsilon\)-NSGAII. The results are shown for 1, 2, 4, 8, and 16 processors. The best overall metrics are shown in underlined bold.

4.4 LTM case study averages (AVG) and standard deviations (STD) in terms of the convergence (Conv.), diversity (Div.), \(\varepsilon\)-indicator (Eind.), and \(\varepsilon\)-performance (Eperf.) metrics. All metrics were computed using 50 random trials. MS and MP designate master-slave and multiple-population versions of the \(\varepsilon\)-NSGAII. The results are shown for 1, 2, 4, 8, and 16 processors. The best overall metrics are shown in underlined bold.

5.1 Summary of sensitivity analysis tools in the study

5.2 Summary of SNOW-17 and SAC-SMA parameters

5.3 PEST sensitivities based on the RMSE measure. Dark gray shading designates highly sensitive parameters defined using a threshold value of 1.0. Light gray designates sensitive parameters defined using a threshold value of 0.01. White cells in the table designate insensitive parameters.

5.4 PEST sensitivities based on the TRMSE measure. Dark gray shading designates highly sensitive parameters defined using a threshold value of 0.1. Light gray designates sensitive parameters defined using a threshold value of 0.001. White cells in the table designate insensitive parameters.

5.5 RSA sensitivities based on the RMSE measure. Dark gray shading designates highly sensitive (HS) parameters. Light gray designates sensitive (S) parameters. White cells in the table designate parameters that are not sensitive (NS).
5.6 RSA sensitivities based on the TRMSE measure. Dark gray shading designates highly sensitive (HS) parameters. Light gray designates sensitive (S) parameters. White cells in the table designate parameters that are not sensitive (NS). ............................... 110

5.7 ANOVA single parameter sensitivities based on the RMSE measure. Dark gray shading designates highly sensitive parameters defined using a threshold F value of 460. Light gray designates sensitive parameters defined using a threshold F value of 4.6. White cells in the table designate insensitive parameters. The values in the brackets provide the 95% confidence interval for the F-values (i.e., the unbracketed value ± the bracketed value yields the confidence interval). ............................... 112

5.8 ANOVA single parameter sensitivities based on the TRMSE measure. Dark gray shading designates highly sensitive parameters defined using a threshold F value of 460. Light gray designates sensitive parameters defined using a threshold F value of 4.6. White cells in the table designate insensitive parameters. The values in the brackets provide the 95% confidence interval for the F-values (i.e., the unbracketed value ± the bracketed value yields the confidence interval). ............................... 112

5.9 Coefficients of determination for the ANOVA model. R1 designates a 1st order ANOVA model that neglects parameter interactions. R2 designates a 2nd order ANOVA model that accounts for pairwise parameter interactions. ............................... 112

5.10 Summations of Sobol’s sensitivity indices for 1st order and 2nd order contributions to model output variance. ............................... 113

5.11 Total order sensitivity indices from Sobol’s method computed using the RMSE measure. Dark gray shading designates highly sensitive parameters defined using a threshold value of 0.1. Light gray designates sensitive parameters defined using a threshold value of 0.01. White cells in the table designate insensitive parameters. The values in the brackets provide the 95% confidence interval for the indices’ values (i.e., the unbracketed value ± the bracketed value yields the confidence interval). ............................... 116
5.12 Total order sensitivity indices from Sobol’s method computed using the TRMSE measure. Dark gray shading designates highly sensitive parameters defined using a threshold value of 0.1. Light gray designates sensitive parameters defined using a threshold value of 0.01. White cells in the table designate insensitive parameters. The values in the brackets provide the 95% confidence interval for the indices’ values (i.e., the unbracketed value ± the bracketed value yields the confidence interval).

5.13 Summary of correlation coefficients from the independent testing of each sensitivity method’s effectiveness.

6.1 Summary of SNOW-17 and SAC-SMA parameters.

6.2 Statistics of the precipitation and streamflow data from year 2001 to year 2003.

6.3 Percentage of time when air temperature is below zero and precipitation is larger than zero. $T_{air}$ denotes air temperature and $P$ defines precipitation. The statistics is based on hourly mean area precipitation and air temperature.

6.4 Annual first order sensitivity indices from Sobol’s method computed using the RMSE measure and 24-hour model time steps. Dark gray shading designates highly sensitive parameters defined using a threshold value of 0.1. Light gray designates sensitive parameters defined using a threshold value of 0.01. White cells in the table designate insensitive parameters. The values in the brackets provide the 95% confidence interval for the indices’ values (i.e., the unbracketed value ± the bracketed value yields the confidence interval).

6.5 Annual total order sensitivity indices from Sobol’s method computed using the RMSE measure and 24-hour model time steps. Dark gray shading designates highly sensitive parameters defined using a threshold value of 0.1. Light gray designates sensitive parameters defined using a threshold value of 0.01. White cells in the table designate insensitive parameters. The values in the brackets provide the 95% confidence interval for the indices’ values (i.e., the unbracketed value ± the bracketed value yields the confidence interval).

6.6 Sobol’s indices for each of the 13 parameters analyzed in the May 2002 and September 2003 events. The first and total order indices for each parameter were computed by summing their individual cell-level indices over the SPKP1 watershed’s model domain.
Acknowledgments

First and foremost, I would like to gratefully acknowledge the academic and financial support of my advisor, Dr. Patrick Reed of the Water Resources Group of Penn State’s Department of Civil and Environmental Engineering. Dr. Reed not only provided a wealth of knowledge and expertise in the area of hydrologic model calibration, long-term groundwater monitoring and evolutionary computing, but also provided a great deal of mentorship throughout these three years. Without Dr. Reed’s supporting role in all aspects, I would not be where I am today. I would also like to thank Dr. Thorsten Wagener, Dr. Christopher Duffy, and Dr. Lin, also here at Penn State, for providing their valuable time and constructive comments on the work contained herein. In addition, I would like to thank Joshua Kollat, Katie van Werkhoven, Mukesh Kumar, Shuangcai Li, Zhenxing Zhang, and Gopal Bhatt who have been supportive colleagues throughout my work. Many thanks should be given to the National Science Foundation for the financial support. I would like to thank my good friends, Yu Qiao, Jiakou Wang, Ting Zhu, Lei Fang, Changying Li, Caiyan Lv, Yi Chen, Gen-Han Wu, especially, Longhua Jiang who have provided constant and generous support during the time I spent at Penn State. In the end, I would like to thank my parents and my sister for their continuous support. I would not be at this point without their help.
Chapter 1

Introduction

Over the past decade the increasing availability of spatially distributed hydrometeorological data (e.g., precipitation, air temperature and soil properties) coupled with advances in computational resources has resulted in increasing interest in the development of spatially distributed hydrological models [e.g. 3–7]. Developers of distributed models seek to better simulate watershed behavior by taking advantage of spatially distributed forcing as well as distributed watershed parameters for a broader array of processes such as surface flow, groundwater flow, sediment transport, solute transport, etc. The increasing complexity of distributed models poses several challenges in terms of (1) their severe computational demands relative to lumped watershed models [8, 9], (2) their potential for over-parameterisation [10], and (3) their high dimensional, nonlinear parametric spaces and structural uncertainties [11].

Because of the uncertainties associated with the data, parameters, and model structure, there is no unique parameter set which is “optimal” for the model to yield simulations which best fit the observations [12]. This makes the problem of calibrating complex hydrologic models difficult since multiple parameter combinations could lead to equivalent model performance. Furthermore, the ill-posed inverse problem [12] could generate multiple optimal sets of parameters by using different model performance measures which evaluate the “goodness-of-fit” [13]. Manual calibration of complex hydrologic models is extremely challenging because of problem complexity as well as time constraints. Consequently, efforts should be made to calibrate hydrologic models using automatic tools that can identify mul-
multiple sets of parameters associated with multiple, conflicting performance metrics such as peak flow prediction accuracy and low flow prediction accuracy.

Multiobjective hydrologic model calibration problems are characterized by their non-linearity, high dimension, multi-modality, concavity, and discontinuity \[14-20\]. These properties make the problems extremely difficult to solve. These problem properties have motivated several prior studies to use heuristic-based optimization, and in particular evolutionary algorithms because they have been shown to work well on nonlinear, nonconvex, and multimodal problems [e.g., \[14, 21\]]. Although a majority of prior studies have focused on highly simplified conceptual rainfall-runoff applications, there are an increasing number of recent studies focusing on developing automatic calibration strategies for distributed hydrologic models \[19, 20, 22, 23\].

Given the increasing computational demands posed by automatic calibration methodologies, it is of paramount importance that optimal search strategies can identify high quality solutions efficiently. There is a need to advance our ability to solve multiobjective hydrologic model calibration problems by comprehensively assessing the relative efficiency, effectiveness, reliability, and ease-of-use of currently available evolutionary multiobjective optimization (EMO) tools. This thesis contributes a comprehensive methodology for assessing EMO tools. The EMO tool assessments were used to develop new multiobjective algorithms for distributed parallel computing environments. Parallel computing is used in this thesis to maximize the performance of EMO algorithms given time and computing constraints. The parallel multiobjective solution tools developed in this research were tested on hydrologic model calibration and groundwater monitoring design applications. These applications have a legacy of multiobjective work and encompass a broad range of problem properties (continuous unconstrained and discrete constrained spaces). This thesis contributes the first parallel EMO study in the water resources area that provides insights into the algorithm’s effectiveness, reliability as well as the impact of problem difficulty for parallelizing water resources applications.

There are two important challenges currently constraining hydrologic model calibration: (1) problem complexity and (2) computational demands. Parallelization helps to address the computational demands posed by hydrologic calibration by improving the efficiency, effectiveness, and robustness of the multiobjective so-
olution tools. In the context of problem complexity, sensitivity analysis is valuable for elucidating parameters’ impacts on a model’s response \[24–30\]. Sensitivity analysis results can be used to decide which parameters should be the focus of model calibration efforts, or even as an analysis tool to test if the model behaves according to its underlying assumptions [e.g., \[31\]]. Ultimately, sensitivity methods should serve as diagnostic tools that help to improve mathematical models and potentially help us to identify where gaps in our knowledge are most severe and are most strongly affecting prediction uncertainty. Efforts should be made to fill the gaps in terms of data collection and understanding of the model. Therefore, this thesis research focuses both on the solution as well as the formulation of multiobjective calibration problems. Problem formulation was addressed by investigating effective sensitivity analysis tools that can be utilized to reduce the size and complexity of multiobjective calibration problems. The goal of sensitivity analysis is to reduce the search space by identifying the key parameters that control a hydrologic model’s responses.

Although there are a variety of sensitivity analysis approaches, very little guidance is available regarding sensitivity method selection. In addition, to date, the computational demands and spatial complexity of distributed hydrologic models have limited our ability to understand their parametric interactions and sensitivities. The limited body of recent literature applying sensitivity analysis to spatially distributed hydrologic models highlights the importance and significant challenges posed by this problem \[9, 11, 30, 32–34\]. Therefore, in this thesis, sensitivity analysis is conducted for both lumped hydrologic models and a distributed hydrologic model. The lumped hydrologic model analysis in this thesis seeks to comprehensively evaluate state-of-the-art sensitivity analysis tools with the goal of clarifying their relative advantages and drawbacks. The comparison study is implemented for lumped models due to their reduced computation demands relative to distributed models. The distributed model sensitivity analysis of this thesis utilizes the most effective sensitivity analysis tool identified via the lumped model sensitivity analysis to characterize the spatial and temporal trends of the Hydrology Laboratory Research Distributed Hydrologic Model (HL-RDHM)’s parametric sensitivities and to identify the major factors that control the model’s behavior.

This thesis research has been divided into four studies. Study 1 aims at com-
prehensively assessing the efficiency, effectiveness, reliability, and ease-of-use of current evolutionary multiobjective optimization (EMO) tools for water resources engineering applications. Study 2 seeks to improve the EMO algorithms’ abilities in terms of their computational cost, their attained solution quality and robustness by using parallel computation. Study 3 focuses on identifying an effective sensitivity analysis tool that can be used to reduce the size of the parameter set that must be considered when evaluating and identifying lumped hydrologic models. Study 4 extends Study 3 by using the best performing global sensitivity analysis tool to analyze a computationally intensive distributed hydrologic model to elucidate its temporal and spatial parametric controls. The EMO and sensitivity analysis tools have been developed to exploit parallel computation.

In this thesis, Chapter 2 provides an overview of prior work in the areas of EMO algorithms, multiobjective calibrations, parallel EMO algorithms, and parametric sensitivity analysis. Chapters 3, 4, 5, and 6 discuss the details of studies 1, 2, 3, and 4 respectively. The contributions and overall conclusions of the thesis are presented in chapter 7. Chapter 8 proposes future work.
2.1 Multiobjective terminology and tools

Evolutionary multiobjective optimization (EMO) algorithms are similar to traditional single objective evolutionary algorithms in that all genetic algorithms search complex problem spaces using a process that is analogous to Darwinian natural selection. Evolutionary algorithms use a population-based search in which high quality solutions are evolved using the three basic operators of (1) selection, (2) mating, and (3) mutation. Analogous to natural systems, selection preferentially samples higher fitness solutions and biases the population to converge to the best solutions. The fitness of each solution is determined by how well it satisfies the specified objectives and constraints of a given application. Mating occurs by combining the decision variables of high quality “parent” solutions to create “child” solutions. The mating operator in combination with selection allows evolutionary algorithms (EAs) to globally search promising regions of a problem space. Lastly, mutation perturbs the decision variables that compose population members. Selection in combination with mutation allows EAs to locally search the problem space near a given solution. The primary difference between EMO algorithms and single objective EAs lies in how fitness is assigned (see [35] for details).

The thesis focuses on assessing and enhancing EMO algorithms’ population-based search, which enables them to evolve entire tradeoff (or Pareto) surfaces within a single run for problems with large decision spaces. These methods can solve highly nonlinear, discrete, and non-convex problems without differentiation.
Additionally, these methods have been demonstrated to be capable of solving high-order Pareto optimization problems (i.e., problems with three or more objectives) \[35, 39–42\]. Schaffer \[43\] developed one of the first EMO algorithms termed the vector evaluated genetic algorithm, which was designed to search decision spaces for the optimal tradeoffs among a vector of objectives. Subsequent innovations in EMO have resulted in a rapidly growing field with a variety of solution methods that have been used successfully in a wide range of applications (as reviewed by \[35, 41, 44, 45\]). These solution methods have garnered increased attention over the past decade and have been applied successfully in a wide variety of environmental and water resources applications \[16, 18–20, 22, 23, 26, 39, 46–55\]. Chapter 3 focuses on assessing and enhancing EMO search to address the current challenges posed by nonlinear, multimodal, and computationally intensive water resources and environmental applications. To clarify the material presented in Chapters 3 and 4, a brief overview of the terminology of evolutionary multiobjective optimization is provided below. A solution \(X^*\) is classified as Pareto optimal when there is no feasible solution \(X\) that will improve some objective values without degrading performance in at least one other objective. More formally, solution \(X^* \in \Omega\) is Pareto optimal if for each \(X \in \Omega\) and \(I = 1, 2, ..., k\), either

\[
f_i(X) \geq f_i(X^*) \quad \forall i \in I
\]

or, there is at least one \(i \in I\) so that

\[
f_i(X^*) < f_i(X)
\]

where \(I\) is a set of integers that range from one to the number of total objectives, \(X\) and \(X^*\) are vectors of decision variables, \(\Omega\) is the decision space, \(k\) is the number of objectives, and \(f\) is the objective function. The definition here is based on the assumption that the optimization problem is formulated to minimize all objective values.

Equations \[2.1\] and \[2.2\] state that a Pareto optimal solution \(X^*\) has at least one smaller (better) objective value compared to any other feasible solution \(X\) in the decision space while performing as well or better than \(X\) in all remaining objectives. In multiobjective optimization, the goal is to identify the set of Pareto
Figure 2.1. Example illustration of the Pareto front for a convex, 2-objective minimization problem. The Pareto front is indicated by the bold curve. The full set of feasible solutions includes the Pareto front and the solutions within the non-bolded curve.

optimal solutions. In the minimization example shown in Figure 2.1, the Pareto front consists of the points in objective space along the curve that is convex with respect to the origin. Each axis represents a different objective function to be minimized. The Pareto front is the mapping of Pareto optimal set from the decision space to the objective space. It represents all of the solutions from the point with minimum $f_1$ to the point with minimum $f_2$. The Pareto front is composed of a set of objective vectors which are not dominated by any other objective vectors in the objective space. It provides the engineer choices of a tradeoff between the two objectives by selecting a point along the Pareto front. In the context of the EMO search process, at each generation of the search process, the current “best-known” solution set is termed the non-dominated set. The nondominated set represents the local set of solutions that have not had their performance exceeded in all objectives by any other solution identified during search. Similar to the Pareto front, the non-dominated front is the mapping of non-dominated set from the decision space to the objective space.

2.2 Multiobjective model calibration

The hydrological behavior of a watershed can be conceptualized as a collection of spatially distributed and highly interrelated water, energy and vegetation processes. Any computer-based model of watershed behavior must, therefore, implement this conceptualization using appropriately coupled systems of parametric
mathematical functions; with parameters allowing for the ability to adapt the model to different (but conceptually similar) watersheds. These parameterizations can be of varying complexity, but are, by definition, much simpler than nature itself. Model parameters therefore often become effective parameters that are related to, but not identical with measurable watershed characteristics and have to be estimated by calibrating the model to observed watershed behavior (e.g. streamflow) to account for this discrepancy. Traditional manual calibration methods use trial-and-error based analyses, which are time consuming and difficult to implement for multiple performance objectives (e.g., capturing high flow, average flow, and low flow simultaneously). There is a large body of recent water resources literature analyzing alternative tools and strategies for automatic calibration using simulation-optimization frameworks [14, 16, 18, 54, 56, 59].

Early studies [14] have highlighted that in the context of optimization, the calibration problem is ill-posed, often highly nonlinear, non-convex, and multimodal (i.e., numerous local optima exist). These problem properties have motivated several prior studies to use heuristic-based optimization, and in particular evolutionary algorithms because they have been shown to work well on nonlinear, nonconvex, and multimodal problems [14, 21, 38]. Advances in computational capabilities have led to more complex hydrologic models often predicting multiple hydrologic fluxes simultaneously (e.g. surface and subsurface flows, energy, etc.). In addition, the use of an identification framework based on a single objective function is based on the erroneous assumption that all the available information regarding one hydrologic variable can be summarized (in a recoverable form) using a single aggregate measure of model performance, leading unavoidably to a loss of information and therefore poor discriminative power [60]. These issues have led to an increasing interest in multi-objective optimization frameworks.

The growing body of research in the area of multiobjective calibration [16, 18, 19, 26, 53, 55, 59] has shown that the multiobjective approach is practical, relatively simple to implement, and can provide insights into parameter uncertainty as well as the limitations of a model [16]. Although a majority of prior studies have focused on conceptual rainfall-runoff applications, there are an increasing number of recent studies focusing on developing multiobjective calibration strategies for distributed hydrologic models [10, 20, 22, 23]. Calibrating a
distributed hydrologic model remains a challenging problem because distributed hydrologic models have more complex structures and significantly larger parameter sets that must be specified. Moreover, distributed models are computationally expensive, causing automatic calibration to be subject to severe computational time constraints. There is also a hidden cost in using evolutionary algorithms for hydrologic model calibration that has not been well addressed in the water resources literature. For increasingly complex models with larger parameter sets a EMO algorithm trial run may take several days or longer. Users must carefully consider how EMO algorithms’ search parameters (i.e., population size, run length, random seed, etc.) impact their performance. Moreover, all of the algorithms perform stochastic searches that can attain significantly different results depending on the seeds specified in their random number generators. When a single EMO trial run takes several days, trial-and-error analysis of the performance impacts of EMO algorithm parameters or running the algorithm for a distribution of random trials can take weeks, months, or even years of computation. The increasing size and complexity of calibration problems being considered within the water resources literature necessitates rapid and reliable search tools (see Chapters 3 and 4).

2.3 Parallel evolutionary multiobjective algorithms

2.3.1 A brief introduction to parallel computing

This section briefly overviews the fundamental concepts of parallel computing (for a more detailed discussion see [61–67]). In parallel computing, a complex task is decomposed into several small subtasks and the subtasks are assigned to different computing resources (e.g. multiple processors in a supercomputer) so that they execute different subtasks concurrently. The history of modern parallel computing can be traced back to the 1950s [67] and at the present time parallel computing has become ubiquitous in industrial and academic applications. Parallel computing enables the solution of complex problems through a divide-and-conquer strategy that can dramatically reduce the wall clock time required to solve applications. The effectiveness of parallel computing is determined by the hardware, algorithm, and
From the hardware point of view, parallel computers can be categorized into shared memory architectures and distributed memory architectures. In the shared memory architecture, every processor has the same global memory address table and can directly access all the memory blocks. On the other hand, distributed memory refers to the memory systems connected by a network, in which every processor can access its own local memory directly and get data from non-local memory through cross-computer communication (e.g., Ethernet connections). The communication is an explicit data exchange between processors termed message passing. Figures 2.2a and 2.2b represent the shared memory and distributed memory architectures, respectively. When developing parallel computing software, two models—OpenMP and Message Passing Interface (MPI) are the most popular programming approaches based on shared memory and distributed memory architectures, respectively. This thesis will utilize MPI because of the framework’s improved portability and scalability for distributed memory computing clusters. The purpose of parallel computing is to attain application speedups, which are generally defined as the ratio of wallclock time of the best available serial program execution to the wall clock time of the parallel program. Scalability refers to a parallel implementation’s capability to demonstrate a proportionate increase in parallel speedup with additional processors. For more details about parallel computing, openMP, and MPI, see [61–67].
2.3.2 Parallel EMO algorithms

EMO algorithms have difficulties in solving computationally intensive water resources and environmental applications because of the computational bottleneck posed by evaluating objective functions using numerically expensive software (e.g., hydrologic models or statistical estimators). Parallel computing is well suited for reducing the total function evaluation time and improving the efficiency of EMO applications. Beyond reducing computing times, parallel computing has been shown to improve the quality of EMO solutions and the reliability of the algorithms themselves [35, 68–71].

There are three main parallel paradigms for EMO algorithms [35, 68, 71]: the master-slave model, the multi-population model, and the diffusion model. These models can be hybridized and their implementations vary significantly in the literature. In Chapter 4, novel versions of the master-slave and multi-population models have been developed and tested for smaller scale clusters (<50 processors). Each of these parallelization schemes are described in more detail in the next two sections.

2.3.2.1 Master-Slave model

The simplest parallel EMO model is the master-slave model. As the name indicates, there is a master processor and one or multiple slave processors. The master processor conducts the evolutionary multiobjective operations (e.g., initial population generation, selection, crossover, and mutation) and is also responsible for assigning solutions to the slaves for objective function evaluations. The master processor can also participate in evaluating solutions. The only task of the slave processors is calculating the objective functions and sending their results back to the master processor. The master processor controls the synchronization of the evolutionary process. The master-slave model is illustrated in Figure 2.3. The master slave model can be easily implemented on both shared memory platforms using openMP and distributed memory platforms using MPI. It should be noted that if the function evaluation time is not significant, the overhead (communication time and idle time) may impair the benefits of parallelization (i.e., no speedup with increasing processors). On the other hand, for the computationally expensive
applications, the overhead is only a minor factor of the time cost and thus the master-slave model is more efficient and scalable.

### 2.3.2.2 Multi-Population model

The multi-population model is often called the island model. The approach utilizes multiple populations distributed on different processors each of which has its own independent search. The multiple populations form a global population conceptually. Figure 2.4 illustrates the multi-population model. Cantu-Paz [72] provides a concise conceptual and theoretical discussion of the challenges users must face when trying to develop efficient multiple population parallelization schemes for evolutionary algorithms. Literature reviews [35, 72] highlight that multiple population schemes have emerged as one of the most popular evolutionary algorithm parallelization strategies because of the emergence of distributed computer clusters and because early studies showed the potential for “superlinear” speed-ups (e.g., an algorithm is 5 times faster using only 4 processors). Some basic terms used in the multi-population EMO algorithms literature are given below:
• Deme—population assigned to each processor

• Epoch—number of generations a deme evolves without communicating with other demes

• Topology—describes “processor connectivity” or the rules for communication between demes

• Migration rate—specifies how many solutions a deme will share with other demes

• Migration frequency—specifies how often migration will occur

• Migration policy—specifies how migrants are selected and which solutions are replaced in receiving demes (e.g., best replaces worst)

Effective design and parameterization of multiple population EAs is extremely challenging and directly impacts the success or failure of an application. It is common practice to judge the efficacy of a parallel application using the concept of speedup. It is very important when judging speedup that solution quality should also be monitored. Monitoring solution quality will ensure that prematurely converged results with small clock times and poor solution quality do not bias speedup assessments. Ideally, the goal of parallelization is to attain “linear speedups” which means that when \( P \) processors are used to solve an application the parallel computing clock time will be equal to \( 1/P \) of the serial execution time (i.e., speedup is equal \( P \) or the number of processors used). In some cases, “super linear speedups” have been documented for very effective parallelization schemes where the speedup factors are greater than the number of processors used to solve the application. Attaining good speedups for a multiple population EA is challenging and related directly to computing hardware, decisions on parameter settings, and the difficulty of the application being solved.

As was theoretically analyzed by Cantu-Paz [72], the total parallel computation time for a multiple population EMO algorithm is the sum of the time required for each deme to evolve and the time spent communicating between processors. The component of parallel computation time spent evolving demes is given by the product of the deme size, the number of generations used, and the average
time required per evaluation. Therefore, minimizing the deme size will help to reduce parallel computation times and enhance speedups. Unfortunately, a small deme or population size will often cause EMO algorithms to have a low reliability [73]. It is also noted that as communication between processors increases there will be degradation in parallel speedups. This degradation causes parallel performance to be “asymptotic” as the number of processors increases, or in simpler terms increasing communication costs by using more processors will result in an upper bound limit where adding more processors will not improve parallel execution time. The speedup asymptote is largely controlled by the ratio of function evaluation time and communication times. It is easier to attain efficient speedups when this ratio is large. As will be introduced in Chapter 4 this thesis presents effective multi-population parallelization strategies that will overcome the design and implementation challenges that are currently limiting the use of distributed computing in multiobjective water resources and environmental applications.

2.4 Parameter sensitivity analysis

Model sensitivity analysis characterizes the impact that changes in model inputs have on the model outputs in a strict sense. Sensitivity measures are determined mathematically, statistically, or even graphically (e.g. scatter plot). There are several prior studies that have broadly reviewed the types of sensitivity analysis methods that exist [74–79]. Any sensitivity analysis approach can be broken up into two components [80]: (1) a strategy for sampling the model parameter space (and/or state variable space), and (2) a numerical or visual measure to quantify the impacts of sampled parameters on the model output of interest. The implementation of these two components varies immensely [e.g., 25, 29, 78, 81–83], and guidance is currently lacking to help modelers decide which approach is best suited to the needs of a particular study. Generally, the approaches can be categorized into two main groups–local methods and global methods [22, 84].

The nominal range and differential analysis methods are two well known local parameter sensitivity analysis methods [76, 78]. Nominal range sensitivity analysis is based on calculation of the percentage change of outputs due to the change of input relative to its baseline (nominal). The percentage change is seen
as the sensitivity of the corresponding input. Differential analysis utilizes partial derivatives of the model outputs with respect to the perturbations of the model input. The derivative values are themselves the metrics of the sensitivity. Further analysis can be conducted by approximating the simulation model using Taylor’s series at the baselines of the inputs. Then the variances of the outputs can be calculated through the Taylor’s series and used as the sensitivity measures [76]. The advantages of nominal range and differential analysis include: straightforward implementation, less computational complexity, and availability of some mature software libraries for derivative calculation (e.g. CVODES [85] and PEST [86]). The major drawback of these methods is their tendency to assess local optima in a model’s response surface.

There are a variety of global sensitivity analysis methods such as regional sensitivity analysis (RSA) [24, 87], variance based methods [74], regression based approaches [88, 89], and Bayesian sensitivity analysis [77]. Global methods attempt to explore the full parameter space within pre-defined feasible parameter ranges. In Chapter 5, a suite of sensitivity methods were tested and their relative benefits and limitations for advancing lumped watershed model identification and evaluation are discussed.

In Chapter 5 the four sensitivity analysis approaches tested include PEST, RSA, analysis of variance (ANOVA), and Sobol’s method. These methods were selected for comparison due to their popularity and their prevalence in a large number of applications [25, 28, 29, 31, 32, 90–96]. The sensitivity analysis methods tested in Chapter 5 range from local to global and capture a broad range of analysis methodologies (differential analysis, RSA, and variance-based analysis). The main characteristics of these four methods and the associated statistical sampling schemes used are discussed in more detail in Chapter 5.
Study 1: Assessing state-of-the-art multiobjective tools for hydrologic model calibration

This chapter is drawn from Y. Tang et al.’s paper published in the journal of Hydrology and Earth System Science [97]. The paper provides a comprehensive assessment of state-of-the-art evolutionary multiobjective optimization (EMO) tools’ relative effectiveness in calibrating hydrologic models. The relative computational efficiency, accuracy, and ease-of-use of the following EMO algorithms are tested: Epsilon Dominance Nondominated Sorted Genetic Algorithm-II ($\varepsilon$-NSGAII), the Multiobjective Shuffled Complex EvolutionMetropolis algorithm (MOSCEM-UA), and the Strength Pareto Evolutionary Algorithm 2 (SPEA2). Three test cases were used to compare the algorithms’ performances: (1) a standardized test function suite from the computer science literature, (2) a benchmark hydrologic calibration test case for the Leaf River near Collins, Mississippi, and (3) a computationally intensive integrated surface-subsurface model application in the Shale Hills watershed in Pennsylvania.
3.1 Introduction

The purpose of this chapter is to comprehensively assess the efficiency, effectiveness, reliability, and ease-of-use of current evolutionary multiobjective optimization (EMO) tools for hydrologic model calibration. The following EMO algorithms are tested: $\varepsilon$-NSGAII [98], MOSCEM-UA [53], and SPEA2 [99]. $\varepsilon$-NSGAII is a new algorithm developed by Kollat and Reed [100] that has been shown to be capable of attaining superior performance relative to other state-of-the-art EMO algorithms, including SPEA2 and $\varepsilon$-NSGAII’s parent algorithm NSGAII developed by Deb et al. [101]. The performance of $\varepsilon$-NSGAII is being tested relative to MOSCEM-UA and SPEA2 because these algorithms provide performance benchmarks within the fields of water resources and computer science, respectively. This study contributes a rigorous statistical assessment of the performances of these three evolutionary multiobjective algorithms using a formal metrics-based methodology.

This study bridges multiobjective calibration hydrologic research where MOSCEM-UA [53] represents a benchmark algorithm and EMO research where SPEA2 [35] is a benchmark algorithm. Three test cases are used to compare the algorithms’ performances. The first test case is composed of a standardized suite of computer science test problems [35, 41, 99], which are used to validate the algorithms’ abilities to perform global search effectively, efficiently, and reliably for a broad range of problem types. This is the first study to test MOSCEM-UA on this suite of problems. The second test case is a benchmark hydrologic calibration problem in which the Sacramento soil moisture accounting model (SAC-SMA) is calibrated for the Leaf River watershed located close to Collins, Mississippi. The Leaf River case study has been used in the development of both single and multiobjective objective calibration tools and specifically MOSCEM-UA [14, 17, 18, 26, 53, 102]. The third test case assesses the algorithms’ performances for a computationally intensive integrated hydrologic model calibration application for the Shale Hills watershed located in the Susquehanna River Basin in north central Pennsylvania. The Shale Hills test case demonstrates the computational challenges posed by the paradigmatic shift in environmental and water resources simulation tools towards highly nonlinear physical models that seek to holistically simulate the water cycle. A challenge and contribution of this work is the develop-
ment of a methodology for comprehensively comparing EMO algorithms that have
different search operators and randomization techniques.

3.2 Evolution-based multiobjective search

This thesis contributes the first comprehensive comparative analysis of these algo-

rithms’ strengths and weaknesses in the context of hydrologic model calibration.
The next sections give a brief overview of each tested algorithm as well as a discus-
sion of their similarities and differences. For detailed descriptions, readers should
reference the algorithms’ original published descriptions [53, 98, 99].

3.2.1 Epsilon Dominance NSGAII (ε-NSGAII)

The ε-NSGAII exploits ε-dominance archiving [40, 103] in combination with au-
tomatic parameterization [104] for the NSGA-II [101] to accomplish the following:
(1) enhance the algorithm’s ability to maintain diverse solutions, (2) automatically
adapt population size commensurate with problem difficulty, and (3) allow users
to sufficiently capture tradeoffs using a minimum number of design evaluations.

A sufficiently quantified trade-off can be defined as a subset of Pareto optimal
solutions that provide an adequate representation of the Pareto frontier that can
be used to inform decision making. Kollat and Reed [98] performed a compre-
hensive comparison of the NSGA-II, SPEA2, and their proposed ε-NSGAII on a
4-objective groundwater monitoring application, where the ε-NSGAII was easier
to use, more reliable, and provided more diverse representations of tradeoffs.

As an extension to NSGA-II [101], ε-NSGAII adds the concepts of ε-dominance
[103], adaptive population sizing, and a self termination scheme to reduce the
need for parameter specification [104]. The values of ε, specified by the users
represent the publishable precision or error tolerances for each objective. A high
precision approximation of the Pareto optimal set can be captured by specifying
very small precision tolerances ε. The goal of employing ε-dominance is to enhance
the coverage of nondominated solutions along the full extent of an application’s
tradeoffs, or in other words, to maintain the diversity of solutions. ε-NSGAII is
binary coded and real coded. In this application, the real coded version of the ε-
NSGAII proposed by Kollat and Reed [98] is employed. The \( \varepsilon \)-NSGAII uses a series of “connected runs” where small populations are exploited to pre-condition search with successively adapted population sizes. Pre-conditioning occurs by injecting current solutions within the epsilon-dominance archive into the initial generations of larger population runs. This scheme bounds the maximum size of the population to four times the number of solutions that exist at the user specified resolution. Theoretically, this approach allows population sizes to increase or decrease, and in the limit when the epsilon dominance archive size stabilizes, the \( \varepsilon \)-NSGAII’s “connected runs” are equivalent to time continuation [105], (i.e., injecting random solutions when search progress slows). For more details about \( \varepsilon \)-dominance or the \( \varepsilon \)-NSGAII, please refer to the following studies ([98, 100, 103, 106, 107]).

There are 4 major parameters that need to be specified for \( \varepsilon \)-NSGAII (1) the probability of mating, (2) the probability of mutation, (3) the maximum run time, and (4) the initial population size. The mating and mutation operators and parameters are discussed in more detail in 3.2.4. The maximum run time is defined as the upper limit on the time the user is willing to invest in search. Although epsilons must be specified for every objective, these values are defined by the properties of the application not the evolutionary algorithm. In any optimization application, it is recommended that the user specify the publishable precision or error tolerances for their objectives to avoid wasting computational resources on unjustifiably precise results.

### 3.2.2 Strength Pareto Evolutionary Algorithm 2 (SPEA2)

SPEA2 represents an improvement from the original Strength Pareto Evolutionary Algorithm [99, 108]. SPEA2 overcomes limitations of the original version of the algorithm by using an improved fitness assignment, bounded archiving, and a comprehensive assessment of diversity using k-means clustering. SPEA2 requires users to specify the upper bound on the number of nondominated solutions that are archived. If the number of non-dominated solutions found by the algorithm is less than the user-specified bound then they are copied to the archive and the best dominated individuals from the previous generation are used to fill up the archive. If the size of non-dominated set is larger than the archive size, a k-means clustering
algorithm comprehensively assesses the distances between archive members. A truncation scheme promotes diversity by iteratively removing the individual that has the minimum distance from its neighboring solutions. The archive update strategy in SPEA2 helps to preserve boundary (outer) solutions and guide the search using solution density information. SPEA2 has 4 primary parameters that control the algorithm’s performance: (1) population size, (2) archive size, (3) the probability of mating, and (4) the probability of mutation. For a more detailed description, see [99, 108].

3.2.3 Multiobjective Shuffled Complex Evolution Metropolis (MOSCEM-UA)

MOSCEM-UA was presented by Vrugt et al. [53]. The algorithm combines a Markov Chain Monte Carlo sampler with the Shuffle Complex Evolutionary algorithm (SCE-UA) algorithm [14], while seeking Pareto optimal solutions using an improved fitness assignment approach based on the original SPEA [108]. It modifies the fitness assignment strategy of SPEA to overcome the drawback that individuals dominated by the same archive members are assigned the same fitness values [53, 99]. MOSCEM-UA combines the complex shuffling of the SCE-UA [14, 109] with the probabilistic covariance-annealing process of the Shuffle Complex Evolution Metropolis-UA algorithm [102]. Firstly, a uniformly distributed initial population is divided into complexes within which parallel sequences are also created after sorting the population based on fitness values. Secondly, the sequences are evolved iteratively towards a multivariate normally distributed set of solutions. The moments (mean and covariance matrix) of the multivariate distribution change dynamically because they are calculated using the information from current evolution stage of sequences and associated complexes. Finally, the complexes are reshuffled before the next sequence of evolution. For a detailed introduction to the algorithm, please refer to the research of Vrugt et al. [53].

Based on the findings of Vrugt et al. [53] and our own analysis, MOSCEM-UA’s performance is most sensitive to two parameters: population size and the number of complexes/sequences. All of the remaining parameters (i.e., reshuffling and scaling) were set to the default values in a C source version of the algorithm.
we received from Vrugt in June 2004.

### 3.2.4 Similarities and difference between the algorithms

$\varepsilon$-NSGAII, SPEA2, and MOSCEM-UA all seek the Pareto optimal set instead of a single solution. Although these algorithms employ different methodologies, ultimately they all seek to balance rapid convergence to the Pareto front with maintaining a diverse set of solutions along the full extent of an application’s tradeoffs. Diversity preservation is also important for limiting premature-convergence to poor approximations of the true Pareto set. The primary factors controlling diversity are population sizing, fitness assignment schemes that account for both Pareto dominance and diversity, and variational operators for generating new solutions in unexplored regions of a problem space.

A key characteristic of $\varepsilon$-NSGAII is the algorithm’s ability to adapt population size commensurate to problem difficulty and promote diversity using “time continuation” (i.e., injecting random solutions when search progress slows). Both SPEA2 and MOSCEM-UA are impacted by population size, but currently trial-and-error analysis is necessary to determine an appropriate search population size. With respect to the fitness assignment, these three algorithms all use the Pareto dominance concept. Both MOSCEM-UA and SPEA2 use the fitness assignment method based on the original fitness assignment approach employed in SPEA. MOSCEM-UA improves the original method by adding Pareto rank when assigning fitness values to dominated individuals in the population. SPEA2 considers both dominated and nondominated individuals as well as their density information when applying fitness assignment. The density function is used to differentiate individuals with the same raw fitness values by calculating the distance from current point being considered to a predefined nearest point \[99\]. $\varepsilon$-NSGAII adopts the $\varepsilon$-dominance grid based approach for fitness assignment and diversity preservation \[103\].

Regarding the whole evolution process, MOSCEM-UA is significantly different from SPEA2 and $\varepsilon$-NSGAII although all of them randomly initialize their search populations. As discussed above, MOSCEM-UA uses the complex shuffling method and the Metropolis-Hastings algorithm to conduct search. Offspring are generated
using a multivariate normal distribution developed utilizing information from the current draw of the parallel sequence within a complex. The acceptance of a new generated candidate solution is decided according to the scaled ratio of candidate solution’s fitness to current draw’s fitness of the sequence. Complex shuffling helps communication between different complexes and promotes solution diversity.

Comparatively, SPEA2 and ε-NSGAII adopt the traditional evolutionary operators (e.g. selection, crossover and mutation) in searching. They both use binary tournament selection, simulated binary crossover (SBX), and polynomial mutation. And both of them maintain external archives which store the best solutions found from the random initial generation to final termination generation. However, these two algorithms are different in many aspects. After population initialization, SPEA2 assigns fitness to each individual in the population and the archive. Non-dominated sorting is conducted on all these individuals and then the nondominated solutions are copied to the archive of next generation. Because the archive is fixed in size, either a truncation scheme must be implemented or the best dominated solutions must be used to fill up the archive. Then binary tournament selection with replacement is applied to select individuals for a mating pool. The new population in SPEA2’s next generation is created by applying crossover and mutation operators to the mating pool. The process is repeated until a user specified termination criterion is met.

ε-NSGAII initiates the search with an arbitrarily small number of individuals (e.g., 10-individuals). Binary tournament selection, SBX crossover, and mutation operators are implemented to generate the first child population. Pareto ranks are assigned to the individuals from the parent and children populations. Solutions are selected preferentially based on their non-domination rank. Crowding distances (i.e., Euclidean norms for measuring distance from neighbor solutions) are used to distinguish between the individuals with the same non-domination rank (i.e., larger crowding distances are picked preferentially to promote diversity). At the end of each generation, the external archive is updated with the ε-non-dominated solutions. The archive size and population size change dynamically based on the total number of ε non-dominated solutions stored. In this study, a single termination criterion based on the maximum number of function evaluations was used for all of the algorithms (i.e., they all had identical numbers of function evaluations) to
ensure a fair comparison.

3.3 Case studies

3.3.1 Case study 1: The test function suite

The first test case is composed of a standardized suite of computer science test problems [35, 41, [110]], which are used to validate the algorithms’ abilities to perform global search effectively, efficiently, and reliably for a broad range of problem types. This is the first study to test MOSCEM-UA on this suite of problems. The test function suite has been developed collaboratively by the EMO community [35, [111] as standardized performance tests where new algorithms must meet or exceed the performance of current benchmark algorithms such as SPEA2.

Since these test functions have been used very broadly in the EMO literature [35, 41, 100, [110]], their detailed formulations will not be presented here. Table 3.1 provides an overview of the number of the decision variables used, their ranges, and the problems’ characteristics. The test functions are labeled $T_1$, $T_2$, $T_3$, $T_4$, and $T_6$ following the naming convention of Zitzler et al. [110]. All of the test functions have been implemented in the standard forms used in the EMO literature. Generally, $T_1$ and $T_2$ are considered relatively straightforward convex and non-convex test problems. $T_3$ tests algorithms’ abilities to find discontinuous convex sets of solutions. $T_4$ and $T_6$ are the most challenging of the test functions requiring algorithms to overcome large numbers of local fronts and non-uniformly distributed solution spaces, respectively.

3.3.2 Case study 2: Leaf River watershed

The Leaf River SAC-SMA test case represents a benchmark problem within the water resources literature, which has been used extensively for developing tools and strategies for improving hydrologic model calibration [14, 17, 18, 26, 53]. Readers interested in the full details of the Leaf River case study’s dataset should refer to earlier works (e.g., [112]). The Leaf River case study used in this paper has been developed based on the original studies used to develop and demonstrate MOSCEM-UA [53, 102]. The Sacramento Soil Moisture Accounting model is a
Table 3.1. Suite of Test Functions

<table>
<thead>
<tr>
<th>Test Functions</th>
<th>Number of Decision Variables and Parameter Ranges</th>
<th>Main Features of the Pareto optimal front</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>$m = 30; [0, 1]$</td>
<td>Convex</td>
</tr>
<tr>
<td>$T_2$</td>
<td>$m = 30; [0, 1]$</td>
<td>Non-Convex counterpart to $T_1$</td>
</tr>
<tr>
<td>$T_3$</td>
<td>$m = 30; [0, 1]$</td>
<td>Discreteness: Multiple non-contiguous convex parts</td>
</tr>
<tr>
<td>$T_4$</td>
<td>$m = 10; [0, 1]$ for the first variable, [-5, 5] for others</td>
<td>Multimodality: $21^9$ local fronts</td>
</tr>
<tr>
<td>$T_6$</td>
<td>$m = 10; [0, 1]$</td>
<td>Solutions are non-uniformly distributed; Solution density is lowest near the front and highest away from the front</td>
</tr>
</tbody>
</table>

16 parameter lumped conceptual watershed model used for operational river forecasting by the National Weather Service throughout the US (see Burnash [113] for more details on the model). All three algorithms searched the same 13 SAC-SMA parameters (3 parameters are commonly fixed a priori) and parameter ranges as were specified by Vrugt et al. [53]. The algorithms were tested on their ability to quantify a 2-objective tradeoff based on a root-mean square error (RMSE) problem formulation. The first objective was the non-transformed RMSE objective, which is largely dominated by peak flow prediction errors due to the use of squared residuals. The best known approximation set generated for this problem is discussed in more detail in the results of this study (see Figure 3.4a). The second objective was formulated using a Box-Cox transformation of the hydrograph ($z = [(y + 1)^\lambda - 1]/\lambda$ where $\lambda = 0.3$) as recommended by [114] to reduce the impacts of heteroscedasticity in the RMSE calculations (also increasing the influence of low flow periods).

A 65-day warm-up period was used in this study based on the methodological recommendations of [53]. A two-year calibration period was used in this study from October 1, 1952 to September 30, 1954. This shortened calibration period was used for this study to control the computational demands posed by rigorously assessing the EMO algorithms. A total of 150 EMO algorithm trial runs were used in this case study (i.e., 50 trials per algorithm). Each EMO algorithm trial run utilized 100,000 SAC-SMA model evaluations, yielding a total of 15,000,000 SAC-SMA model evaluations used in our Leaf River case study analysis. Reducing the
calibration period improved the computational tractability of our analysis. The focus of this study is on assessing the performances of the three EMO algorithms which are captured in the 2 year calibration period. In actual operational use of the SAC-SMA for the Leaf River 8 to 10 year calibration periods are used [18].

3.3.3 Case study 3: Shale Hills watershed

The Shale Hills experimental watershed was established in 1961 and is located in the north of Huntington County, Pennsylvania. It is located within the Valley and Ridge province of the Susquehanna River Basin in north central Pennsylvania. The data used in this study was supplied by a comprehensive hydrologic experiment conducted in 1970 on a 19.8 acre sub-watershed of the Shale Hill experimental site. The experiment was led by Jim Lynch of the Pennsylvania State University’s Forestry group with the purpose of exploring the physical mechanisms of the formation of stream-flow at the upland forested catchment and to evaluate the impacts of antecedent soil moisture on both the volume and timing of the runoff (see [115]). The experiment was composed of an extensive below canopy irrigation network for simulating rainfall events as well as a comprehensive piezometer network, 40 soil moisture neutron access tubes and 4 weirs for measuring flow in the ephemeral channel. Parameterization of the integrated model for the Shale Hills was also supported by more recent site investigations, where Lin et al. [116] extensively characterized the soil and groundwater properties of the site using in-situ observations and ground penetrating radar investigations.

3.3.3.1 Integrated model description

The hydrologic model being calibrated in this study is a semi-distributed version of the integrated hydrologic model being developed by Duffy et al. [7, 115, 117]. This model integrates watershed processes within the terrestrial hydrologic cycle over a wide range of time scales. It couples surface, subsurface and channel states within the hillslope and watershed. The model strategy is to transform partial differential equations (PDEs) to ordinary differential equations (ODEs), using the semi-discrete finite volume method (SD-FVM) [2]. Specifically, the spatial domain is decomposed into different zones (response units). Different ODEs are
created to simulate different hydrologic processes within each zone. The ODE system within each zone is termed a “Model Kernel”. An overall ODE system is created by combining all of the model kernels. The ODE system is solved using an implicit Runge-Kutta ODE solver (RADAU IIA) of order 5. As noted by Duffy, by taking advantage of the finite volume method, the model strategy has the capability of capturing the “dynamics” in different processes while maintaining the water balance. The model also has the flexibility of easily adding/eliminating (switching on/off) the key hydrologic processes for a system.

As discussed above, the water budget is computed using a global model kernel composed of ODEs representing each of the watershed zones or river sections. The number of ODEs increases linearly with the number of decomposed spatial zones within the watershed. In the Shale Hills application, the watershed is decomposed into 7 zones and 4 river sections connected to each other between the zones. The decomposed domain and the topology of the zones and the river sections are shown in Figure 3.1. The domain decomposition results in 32 ODEs solved implicitly using a solver that has been proven to be highly effective for ODE systems. The model simulation time is substantial for this study given that the EMO algorithms will have to evaluate thousands of simulations while automatically calibrating model parameters. On a Pentium 4 Linux workstation with a 3 gigahertz processor and 2 gigabytes of RAM, a one month simulation of Shale Hills using a 1 hour output time interval requires 120 seconds of computing time. If 5000 model evaluations are used to optimize model parameters, then a single EMO run will take almost 7 days. This study highlights how trial-and-error analysis of EMO algorithm performance can have a tremendous cost in both user and computational time.

3.3.3.2 Problem formulation

Multiobjective calibration uses multiple performance measures to improve model predictions of distinctly different responses within a watershed’s hydrograph simultaneously (e.g., high flow, low flow, average flow). For the Shale Hills case study, the calibration objectives were formulated to generate alternative model parameter groups that capture high flow, average flow, and low flow conditions for the Shale Hills test case using the three search objectives given in equations 3.1 – 3.3. The
Figure 3.1. Domain decomposition of the Shale Hills test case.

The problem formulations used in this study build on prior research using RMSE and the heteroscedastic maximum likelihood estimator (HMLE) measures [15–20, 56].

\[ f_1(\theta) = \left( \frac{1}{N} \sum_{i=1}^{N} W_{1i}[Q_{\text{obs},i} - Q_{\text{sim},i}(\theta)]^2 \right)^{1/2} \]  
\[ f_2(\theta) = \left[ \frac{1}{\sum_{j=1}^{M_p} \sum_{j=1}^{n_j} w_{2i} [Q_{\text{obs},i} - Q_{\text{sim},i}(\theta)]^2} \right]^{1/2} \]  
\[ f_3(\theta) = \left[ \frac{1}{\sum_{j=1}^{M_l} \sum_{j=1}^{n_j} w_{3i} [Q_{\text{obs},i} - Q_{\text{sim},i}(\theta)]^2} \right]^{1/2} \]

Where \( Q_{\text{obs},i} \) is the observed discharge at time \( i \); \( Q_{\text{sim},i} \) is the simulated discharge; \( N \) is the total number of time steps in the calibration period; \( M_p \) is the number of peak flow events; \( M_l \) is the number of low flow events; \( n_j \) is the number of time steps in peak/low flow event number \( j \); \( w_{1i} \), \( w_{2i} \), and \( w_{3i} \) are the weighting functions; \( \theta \) is the set of model parameters to be calibrated. In this study, the weighting coefficients for high flow and low flow are adapted forms of HMLE statistics [56]. The weights for high flow errors are set to the square of the observed discharges to emphasize peak discharge values. The weights for low flow are set to give prominence to low flow impacts on the estimation errors. The weighting coefficient for average flow is set to 1 and thus the error metric for average flow is the standard RMSE statistic. Equation 3.4 provides the weighting coefficients.
Table 3.2. Parameters being optimized in the Shale Hills case study.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Description</th>
<th>Units</th>
<th>Min.</th>
<th>Max.</th>
<th>Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_s$</td>
<td>Saturated hydraulic conductivity</td>
<td>m/h</td>
<td>0.000035</td>
<td>0.15</td>
<td>Zone</td>
</tr>
<tr>
<td>$\eta$</td>
<td>Surface time scale</td>
<td>1/h</td>
<td>0.08</td>
<td>1</td>
<td>Zone</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Empirical constant</td>
<td>1/m</td>
<td>0</td>
<td>7</td>
<td>Zone</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Empirical constant</td>
<td></td>
<td>1.1</td>
<td>2</td>
<td>Zone</td>
</tr>
<tr>
<td>$n$</td>
<td>Manning’s coefficient</td>
<td></td>
<td>0.02</td>
<td>0.08</td>
<td>River Section</td>
</tr>
<tr>
<td>$K_{sr}$</td>
<td>Saturated hydraulic conductivity of river section</td>
<td>m/h</td>
<td>0.000035</td>
<td>0.3</td>
<td>River Section</td>
</tr>
</tbody>
</table>

used to differentiate different hydrologic responses.

\[
\begin{align*}
    w_1 &= 1 \\
    w_2 &= Q_{obs}^2 \\
    w_3 &= \left( \frac{1}{Q_{obs}^2} \right)^{1/\sum_{j=1}^{M} n_j}
\end{align*}
\]  

Preliminary sensitivity analysis showed that the model was very sensitive to the initial surface storage, but the impacts of the initial surface storage were attenuated within the first 100 hours. Figure 3.2 illustrates the Shale Hills calibration period including a 100 hour warm up period to reduce the impacts of the initial conditions. High flow and low flow classifications were made based on points of inflection within the hydrograph. Table 3.2 overviews the parameters being calibrated for the Shale Hills case study. For overland flow the convergence time scale of a hill slope can not be estimated analytically so the parameter was selected for calibration. The saturated soil hydraulic conductivity is calibrated as well as the empirical constants ($\alpha$, $\beta$) in the van Genuchten soil functions. In our preliminary sensitivity analysis, Manning’s coefficient ($n$) and the saturated hydraulic conductivity of river reaches were identified to significantly impact river routing and groundwater-stream interactions. Both of these parameters are calibrated. In the Shale Hills case study, a total of 36 parameters are being calibrated (7 spatial zones * 4 parameters + 4 river sections * 2 parameters). The parameter ranges were specified based on both field surveys \cite{116, 117} and recommendations from literature \cite{120, 121}.
Figure 3.2. Illustration of the Shale Hills calibration period where a 100 hour warm up period was used. High flow and low flow classifications were made based on the points of inflection within the hydrograph.

### 3.4 Description of computational experiment

#### 3.4.1 Algorithm configurations and parameterizations

In an effort to ensure a fair comparison between $\epsilon$-NSGAII and each of the other algorithms, significant effort has been focused on seeking optimal configurations and parameterizations for SPEA2 and MOSCEM-UA using trial-and-error analysis and prior literature. The broadest analysis of the impacts of alternative algorithm configurations was done for the test function suite, since this test case has the smallest computational demands. The algorithms were allotted 15,000 function evaluations for each trial run when solving each problem within the test function suite based on the recommendations and results of prior studies [99, 100]. For each problem in the test function suite a total of 350 trial runs were performed (i.e., 1 configuration for $\epsilon$-NSGAII tested for 50 random seeds, 4 MOSCEM-UA configurations tested for 50 random seeds each yielding 200 trial runs, and 2 SPEA2 configurations tested for 50 random seeds each yielding 100 trial runs).

Since $\epsilon$-NSGAII and SPEA2 use the same mating and mutation operators, the algorithms’ probabilities of mating where set equal 1.0 and their probabilities of mutation were set equal to $1/L$ where $L$ is number of decision variables as has been recommended extensively in the literature [41, 101, 110]. $\epsilon$-NSGAII
utilized an initial population size of 10 individuals. For the test function suite SPEA2’s two configurations both used an archive size of 100 based on prior studies \[35, 41, 99, 110\] and two different population sizes \(N = 100\) and \(N = 250\). MOSCEM-UA’s configurations tested the impacts of increasing population sizes \(N\) and increasing the numbers of complexes \(C\): \((N = 100, C = 2)\), \((N = 250, C = 2)\), \((N = 250, C = 5)\) and \((N = 1000, C = 5)\). The largest population size and number of complexes tested for MOSCEM-UA were based on a personal communication with the Jasper Vrugt, the algorithm’s creator.

\(\varepsilon\)-NSGAII utilized the same configuration as was used for the test function suite on the Leaf River and Shale Hills case studies in an effort to test the algorithms’ robustness in the absence of trial-and-error analysis. Based on the SPEA2’s performance on the test function suite and trial-and-error analysis the algorithm’s population size was set equal to 100 for both the Leaf River and Shale Hills test cases. A key challenge in maximizing the performance of SPEA2 lies in specifying an effective archive size without a priori knowledge of the Pareto set. SPEA2’s performance is very sensitive to archive size. Trial-and-error analysis revealed that if the algorithm’s archive is too small than its overall performance suffered. Moreover, setting the SPEA2 archive to be very large also reduced the algorithm’s search effectiveness because its diversity enhancing clustering operator is under utilized. For the Leaf River and Shale Hills case studies, SPEA2’s performance was maximized by setting the archive size equal to 500 and 100, respectively, based on the average archive sizes attained by the \(\varepsilon\)-NSGAII. Note \(\varepsilon\)-NSGAII automatically sizes its archive based on the number of \(\varepsilon\)-nondominated solutions that have been found.

For the Leaf River case study, MOSCEM-UA utilized a population size of 500 individuals and 10 complexes as was used by Vrugt et al. \[53\] in the original development and demonstration of the algorithm. As will be discussed in the results presented in Section 5 increasing the population size and number of complexes used by MOSCEM-UA has a very large impact on the algorithm’s solution time, which significantly impacted our analysis of the Shale Hills test case. For the Shale Hills case study, MOSCEM-UA was tested for a population size of 250 with 2 or 5 complexes to ensure that a single run would complete in 7 days based on the maximum runtimes we were allotted for the LION-XO computing cluster. The
computational constraints limiting our ability to use larger population sizes and more complexes in the Shale Hills trial runs for MOSCEM-UA are discussed in greater detail in Section 3.5.

3.4.2 Performance metrics

The performances of all of the EMO algorithms tested in this study were assessed using metrics designed to answer two questions: (1) how good are the approximation sets found by the EMO algorithms? and (2) which of the solution sets are better than the others? Deb and Jain \([122]\), stress that EMO performance assessments must account for two separate and often conflicting approximation set properties: (1) convergence—the distance from the reference set of optimal solutions, and (2) diversity—how well the evolved set of solutions represents the full extent of the tradeoffs that exist between an application’s objectives. Performance metrics that measure these properties are termed unary indicators because their values are calculated using one solution set and they reveal specific aspects of solution quality \([123]\).

Two unary metrics, the \(\varepsilon\)-indicator \([123]\) and the hypervolume indicator \([108]\) were selected to assess the performances of the algorithms. The unary \(\varepsilon\)-indicator measures how well the algorithms converge to the true Pareto set or the best known approximation to the Pareto set. The unary \(\varepsilon\)-indicator represents the smallest distance that an approximation set must be translated to dominate the reference set, so smaller indicator values are preferred. The unary hypervolume metric measures how well the algorithms performed in identifying solutions along the full extent of the Pareto surface or its best known approximation (i.e., solution diversity). The unary hypervolume metric was computed as the difference between the volume of the objective space dominated by the true Pareto set and volume of the objective space dominated by the approximation set. Ideally, the hypervolume metric should be equal to zero. For more details about the descriptions and usages of these metrics, see \([98, 108, 123]\).

In addition to the unary metrics discussed above, performance was also assessed using a binary metric. The binary metric was implemented by combining the unary \(\varepsilon\)-indicator metric with an interpretation function. Zitzler et al. \([123]\) formulated
the interpretation function to directly compare two approximation solution sets and conclude which set is better or if they are incomparable. The term “binary” refers to the metric’s emphasis on comparing the quality of two approximation sets. The $\varepsilon$-indicator and the interpretation function are formulated as shown in equations (3.5) and (3.6) separately:

$$I_\varepsilon(A, B) = \max_{z^2 \in B} \min_{z^1 \in A} \max_{1 \leq i \leq n} \frac{z^1_i}{z^2_i}$$

(3.5)

$$F = (I_\varepsilon(A, B) \leq 1 \land I_\varepsilon(B, A) > 1)$$

(3.6)

Where $z^1 \in A$ and $z^2 \in B$ are objective vectors; $A$ and $B$ are two approximation sets; $F$ is an interpretation function. If $A$ is not better than $B$ and $B$ is not better than $A$, then the sets are incomparable. When $F$ is true, it indicates that is $A$ better than $B$. Similarly, changing the order of $A$ and $B$, the decision about whether $B$ is better than $A$ can be made.

The binary $\varepsilon$-indicator metric provides a direct way of ranking the quality of approximation sets generated using different initial random populations and/or different algorithm configurations. The results of each trial run are compared to the results of all other trial runs in the comparison pool. Each trial run is given a rank according to the number of trial runs that exceed its performance in terms of the binary $\varepsilon$-indicator metric. The best trial runs are assigned a rank of one, while a rank of two is assigned to the trial runs that have the second best results. The process is repeated until every trial run is assigned a rank. The trial runs in the same rank are incomparable to one another. In this study, the binary $\varepsilon$-indicator ranking results are presented in terms of the ratio of trial runs that attain top ranks (i.e., ranks of 1 or 2).

3.5 Results

3.5.1 Optimization results for the test function suite

As described in Section 3.4.2 the binary $\varepsilon$-indicator metric provides performance rankings for alternative algorithm configurations and cross-algorithm performance. For each test problem a total of 350 trial runs were performed (i.e., 1 configuration
Table 3.3. Test function results for the ratios of top trial runs for each configuration of the algorithms based on the binary $\varepsilon$-indicator metric ranking. The values highlighted by bold font are the best values among the configurations within a specific algorithm, the values indicated by bold font with underscore are the best values across algorithms.

<table>
<thead>
<tr>
<th>MOEA Configurations</th>
<th>Top Ranking Ratios</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$T_1$</td>
</tr>
<tr>
<td>$\varepsilon$-NSGAII (N=10)</td>
<td>50/50</td>
</tr>
<tr>
<td>SPEA2 (N=100)</td>
<td>50/50</td>
</tr>
<tr>
<td>(N=250)</td>
<td>50/50</td>
</tr>
<tr>
<td>MOSCEM-UA</td>
<td></td>
</tr>
<tr>
<td>(N=100,C=2)</td>
<td>0/50</td>
</tr>
<tr>
<td>(N=250,C=2)</td>
<td>1/50</td>
</tr>
<tr>
<td>(N=250,C=5)</td>
<td>0/50</td>
</tr>
<tr>
<td>(N=1000,C=5)</td>
<td>11/50</td>
</tr>
</tbody>
</table>

For $\varepsilon$-NSGAII tested for 50 random seeds, 4 MOSCEM-UA configurations yielding 200 random seed trials, and 2 SPEA2 configurations yielding 100 random seed trials. After ranking the trial runs, we present the ratio of the number of top ranking runs out of the 50 trials used to test each of the algorithms’ configurations (see Table 3.3). The best configurations for SPEA2 and MOSCEM-UA are and , respectively. The $\varepsilon$-NSGAII has the best overall binary $\varepsilon$-indicator metric rankings for the test function suite.

The unary hypervolume and $\varepsilon$-indicator metrics measure solution diversity and algorithm convergence to the true Pareto fronts, respectively. These unary metrics provide a more detailed understanding of the dynamic performances of the algorithms in terms of efficiency, effectiveness, and reliability. The means and standard deviations of the final optimization results for the best configurations ($\varepsilon$-NSGAII has only one configuration) are summarized in Table 3.4. Recall that the unary $\varepsilon$-indicator represents the smallest distance that an approximation set must be translated to dominate the reference set so smaller indicator values are preferred. Likewise, the unary hypervolume metric is the difference between the volume of the objective space dominated by the true Pareto set and volume of the objective space dominated by the approximation set. Ideally, the hypervolume metric should be equal to zero.

In Table 3.4 the $\varepsilon$-NSGAII has the best overall average performance in both metrics for the test functions. In addition, the relatively small standard deviations reveal that $\varepsilon$-NSGAII is reliable in solving the test functions. SPEA2 is also
Table 3.4. Averages and standard deviations of the unary metrics for each algorithm’s best configuration. AVG stands for mean, STD stands for standard deviation, and bolded entries highlight the best value attained.

<table>
<thead>
<tr>
<th>MOEA</th>
<th>Hypervolume (AVG, STD)</th>
<th>ε-Indicator (AVG, STD)</th>
<th>Time (s) (AVG, STD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_1$</td>
<td>ε-NSGAII</td>
<td>0.000143, 0.000075</td>
<td>0.004119, 0.002017</td>
</tr>
<tr>
<td></td>
<td>SPEA2</td>
<td>0.013144, 0.002119</td>
<td>0.016089, 0.002391</td>
</tr>
<tr>
<td>MOSCEM-UA</td>
<td>0.669011, 0.473327</td>
<td>0.362347, 0.196693</td>
<td>397.42, 165.54</td>
</tr>
<tr>
<td>$T_2$</td>
<td>ε-NSGAII</td>
<td>0.000291, 0.001579</td>
<td>0.009849, 0.022170</td>
</tr>
<tr>
<td></td>
<td>SPEA2</td>
<td>0.530147, 0.163352</td>
<td>0.530147, 0.495096</td>
</tr>
<tr>
<td>MOSCEM-UA</td>
<td>0.510789, 0.222421</td>
<td>0.460137, 0.180662</td>
<td>296.24, 18.45</td>
</tr>
<tr>
<td>$T_3$</td>
<td>ε-NSGAII</td>
<td>0.037842, 0.055236</td>
<td>0.170834, 0.210332</td>
</tr>
<tr>
<td></td>
<td>SPEA2</td>
<td>0.026059, 0.009598</td>
<td>0.030789, 0.022817</td>
</tr>
<tr>
<td>MOSCEM-UA</td>
<td>1.075623, 0.430695</td>
<td>0.750717, 0.348724</td>
<td>307.34, 22.14</td>
</tr>
<tr>
<td>$T_4$</td>
<td>ε-NSGAII</td>
<td>0.017256, 0.042293</td>
<td>0.023307, 0.048349</td>
</tr>
<tr>
<td></td>
<td>SPEA2</td>
<td>1.652341, 0.606150</td>
<td>1.934984, 0.659157</td>
</tr>
<tr>
<td>MOSCEM-UA</td>
<td>51.030919, 6.691913</td>
<td>49.399170, 7.289342</td>
<td>732.71, 89.64</td>
</tr>
<tr>
<td>$T_6$</td>
<td>ε-NSGAII</td>
<td>0.015059, 0.001569</td>
<td>0.280775, 0.000000</td>
</tr>
<tr>
<td></td>
<td>SPEA2</td>
<td>0.042271, 0.004423</td>
<td>0.042271, 0.000000</td>
</tr>
<tr>
<td>MOSCEM-UA</td>
<td>1.478574, 1.066505</td>
<td>0.784151, 0.339040</td>
<td>551.93, 167.39</td>
</tr>
</tbody>
</table>

effective and reliable in solving the test functions. Both $\varepsilon$-NSGAII and SPEA2 are superior to MOSCEM-UA. Figure 3.3 illustrates the variability in the algorithms’ performances by presenting runtime results for the $\varepsilon$-indicator distance metric. The plots show the results of all 50 random seed trials with the mean performance indicated by a solid line, the standard deviation by a dashed line, and the range of random seed performance indicated by the shaded region. Visualizing the results in this manner allows for comparison between the dynamics and reliability (i.e., larger shaded regions indicate lower random seed reliability) of each algorithm.

Figure 3.3 confirms that $\varepsilon$-NSGAII was both the most efficient and effective of the algorithms attaining very close approximations of the true Pareto sets in under 2500 evaluations. SPEA2 typically requires 7500 evaluations to attain equivalent metric values relative to $\varepsilon$-NSGAII. MOSCEM is the least reliable and efficient of the algorithms for the test function suite, failing to attain competitive results in 15,000 evaluations. Dynamic plots of the hypervolume metric showed very similar results to the runtime unary $\varepsilon$-indicator results shown in Figure 3.3. The most significant performance differences between the algorithms resulted for the multimodal $T_4$ problem. The performance rankings in Table 3.3 show that MOSCEM-
Figure 3.3. Dynamic performance plot for the unary $\varepsilon$-indicator distance metric versus total design evaluations for the best performing configurations of the $\varepsilon$-NSGAII, SPEA2, and MOSCEM. Mean performance is indicated by a solid line, the standard deviation by a dashed line, and the range of performance by the shaded region. The plots were generated using 50 trials for each algorithm.

UA generally failed to converge to the Pareto front for $T_4$. SPEA2’s dynamic search results for $T_4$ (see Figure 3.3) are much better than MOSCEM-UA but its final solution set is still far away from the Pareto front as evidenced by its poor ranking results in Table 3.3. Only $\varepsilon$-NSGAII successfully converges to the true Pareto front for $T_4$ with high reliability. In terms of elapsed computational time, the $\varepsilon$-NSGAII is an order of magnitude faster than that of SPEA2, and the elapsed computational time of SPEA2 is an order of magnitude faster than MOSCEM-UA. For example, in solving $T_1$, the average computational times required by $\varepsilon$-NSGAII, SPEA2 and MOSCEM-UA are 1.90 seconds, 21.75 seconds, and 397.42
seconds, respectively. Note this difference in computational efficiency had dramatic impacts on the computational times required for our test function analysis, where several days were required for MOSCEM-UA, several hours for SPEA2, and several minutes for $\varepsilon$-NSGAII.

Averaged performance metrics are meaningful only in cases when the EMO algorithms’ metric distributions are significantly different from one another. In this study, the Mann-Whitney test [124] was used to validate that the algorithms attained statistically significant performance differences. The null hypothesis for the tests assumed that metric distributions for any two algorithms are the same. The Mann-Whitney test showed a greater than 99% confidence that performance metric scores for the $\varepsilon$-NSGAII are significantly different those of MOSCEM-UA for all of the test functions. When comparing SPEA2 and MOSCEM-UA it was found that the algorithms’ performance differences on $T_2$ are not statistically significant. On all of the remaining test functions SPEA2’s superior performance relative to MOSCEM-UA was validated at greater than a 99% confidence level. The $\varepsilon$-NSGAII’s performance was statistically superior to SPEA2 at the 99% confidence level for all of the test functions except for $T_3$. $\varepsilon$-NSGAII and SPEA2 did not attain a statistically meaningful performance difference on $T_3$.

3.5.2 Optimization results for the Leaf River case study

The performance metrics utilized in this study require a reference Pareto set or the best known approximation to the Pareto optimal set. The best known approximation set was generated by collecting all of the nondominated solutions generated from the 150 trial runs used for this case study (i.e., 50 trial runs per algorithm). Figure 3.4 shows the 2-objective tradeoff between low flow RMSE and peak flow RMSE. $\varepsilon$-NSGAII found 58% of the reference set and the remaining 42% of the reference set was generated by SPEA2. MOSCEM-UA was unable to contribute to the best solutions that compose the reference set. Figure 3.5 demonstrates the simulated hydrographs using the parameters values associated the reference set. Table 3.5 shows that SPEA2 was able to attain the best binary $\varepsilon$-indicator metric rankings followed by $\varepsilon$-NSGAII and lastly MOSCEM-UA. Table 3.6 shows that SPEA2 had the best average performance in terms of both the $\varepsilon$-indicator and
Figure 3.4. (a) Reference set generated for the Leaf River test case (b) the percentage of the reference set contributed by $\varepsilon$-NSGAII, SPEA2, and MOSCEM-UA.

Table 3.5. Leaf River case study’s ratios of top trial runs for each configuration of the algorithms based on the binary $\varepsilon$-indicator metric ranking. The best performing algorithm is highlighted in bold.

<table>
<thead>
<tr>
<th>MOEA Configurations</th>
<th>Top Ranking Ratios</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$-NSGAII (N=10)</td>
<td>13/50</td>
</tr>
<tr>
<td>SPEA2 (N=100)</td>
<td>42/50</td>
</tr>
<tr>
<td>MOSCEM (N=500, C=10)</td>
<td>13/50</td>
</tr>
</tbody>
</table>

hypervolume unary metrics. The Mann-Whitney test validated that SPEA2’s results were different from both MOSCEM-UA and $\varepsilon$-NSGAII at the 99% confidence level.

The results of Table 3.6 demonstrate that average performance metrics can be misleading without statistical testing. Although MOSCEM-UA has superior mean hypervolume and $\varepsilon$-indicator distance values relative to $\varepsilon$-NSGAII, performance differences between the algorithms were not statistically significant (i.e., the null hypothesis in the Mann-Whitney test could not be rejected). In fact, all three

Table 3.6. Leaf River case study’s results for the averages and standard deviations of the unary metrics for each algorithm configuration. AVG stands for mean, STD stands for standard deviation, and bolded entries highlight the best value attained.

<table>
<thead>
<tr>
<th>MOEA</th>
<th>Hypervolume AVG</th>
<th>Hypervolume STD</th>
<th>$\varepsilon$-Indicator AVG</th>
<th>$\varepsilon$-Indicator STD</th>
<th>Time(s) AVG</th>
<th>Time(s) STD</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$-NSGAII</td>
<td>1.106633</td>
<td>1.040681</td>
<td>0.530667</td>
<td>0.477824</td>
<td>828.62</td>
<td>35.78</td>
</tr>
<tr>
<td>SPEA2</td>
<td><strong>0.296110</strong></td>
<td>0.432144</td>
<td><strong>0.138678</strong></td>
<td>0.230176</td>
<td>833.08</td>
<td>18.16</td>
</tr>
<tr>
<td>MOSCEM-UA</td>
<td>0.548759</td>
<td>0.649281</td>
<td>0.304506</td>
<td>0.334294</td>
<td>1237.91</td>
<td>59.52</td>
</tr>
</tbody>
</table>
Figure 3.5. Hydrograph ranges corresponding to reference set of the Leaf River test Case. The shaded area between dashed lines is the range of simulated hydrographs associated with the parameters groups in reference set. The bold line indicates the observed streamflow.

algorithms had significant ranges of performance for this test case because of the presence of a large false front (i.e., the locally nondominated front shown in Figure 3.4), which caused some of the algorithms' runs to miss the best known front. Figure 3.6 illustrates the variability in the algorithms' performances by presenting runtime results for the $\varepsilon$-indicator distance metric. Figure 3.6 verifies that SPEA2 has the best mean performance over the full duration of the run. The figure shows that SPEA2 was slightly more reliable relative to $\varepsilon$-NSGAIi and MOSCEM-UA and also verifies that MOSCEM-UA and $\varepsilon$-NSGAIi have nearly identical runtime performance distributions.

Dynamic plots for hypervolume showed similar runtime distributions for the three algorithms. Figure 3.7 illustrates dynamic results for the best performing
Figure 3.6. Leaf River test case dynamic performance results for the unary $\varepsilon$-indicator distance metric versus total design evaluations. Mean performance is indicated by a solid line, the standard deviation by a dashed line, and the range of performance by the shaded region. The plots were generated using 50 trial runs for each algorithm.

Figure 3.7. Dynamic performance plots showing the best performing Leaf River trial runs for each algorithm.

trial runs for each of the algorithms. The plot shows that $\varepsilon$-NSGAII is able to attain superior hypervolume (diversity) and $\varepsilon$-indicator distance (convergence) metrics in less than 5000 model evaluations. SPEA2 and MOSCEM-UA required between 12,000 and 25,000 model evaluations to attain equivalent performance metric values. Overall SPEA2 had superior performance for this test case while MOSCEM-UA and $\varepsilon$-NSGAII had comparable performances.
3.5.3 Optimization results for the Shale Hills test case

For the Shale Hills test case, MOSCEM-UA’s parameters were challenging to set given the computational expense of the integrated hydrologic model. As discussed in Section 3.3.3.1, the Shale Hills test case poses a tremendous computational challenge where a single algorithm trial run requires approximately a week of computation. Given the magnitude of simulation evaluation times, the computational time spent in algorithmic search for both $\varepsilon$-NSGAII and SPEA2 is negligible. Unfortunately, MOSCEM-UA’s algorithmic time is not negligible for increasing population sizes and increasing numbers of complexes because the algorithm utilizes a matrix inversion as part of its stochastic search operators. The severity of MOSCEM-UA’s algorithmic inefficiency is highlighted in the test function analysis where $\varepsilon$-NSGAII was able to solve the test function suite for 50 random seeds in times on the order of minutes whereas MOSCEM-UA required days for population sizes greater than 250. For the Shale Hills case study, MOSCEM-UA was tested for a population size of 250 with 2 or 5 complexes because increasing these parameters caused a single run to exceed the 7 day maximum runtimes we were allotted for the LION-XO computing cluster. The severe computational demands of this test case required that we assess the algorithms using 15 random seed trials. If the 60 trial runs (i.e., 4 algorithm configurations * 15 random seed trials) were run on a single Pentium 4 Linux workstation with a 3 gigahertz processor and 2 gigabytes of RAM this test case would have required approximately 420 days of continuous computation.

The best known approximation set was generated by collecting the nondominated solutions from the 60 trial runs used for this case study. Figure 3.8a shows the best known solution set in the 3-objective solution space defined for this test case. Figure 3.8b projects the solution set onto 2-objective planes to better illustrate the tradeoffs that exist between low, average, and peak flow calibration errors. Figure 3.9 shows that $\varepsilon$-NSGAII found 94% of the reference set and the remaining 6% of the reference set was generated by SPEA2. MOSCEM did not contribute to the best solutions that compose the reference set. The simulated hydrographs using the parameters values corresponding to the reference set are presented in Figure 3.10. Table 3.7 shows that SPEA2 was able to attain slightly better binary $\varepsilon$-indicator metric rankings relative to the $\varepsilon$-NSGAII. As indicated by Figure 3.9 and Table 3.7 MOSCEM had difficulty in generating highly ranked
Figure 3.8. (a) Reference set for the Shale Hills test case (b) projections of the reference set onto the 2-objective planes to highlight the tradeoffs between the objectives.

Figure 3.9. The percentages of the Shale Hills reference set contributed by $\varepsilon$-NSGAII, SPEA2, and MOSCEM-UA.

runs for this test case. Although Table 3.8 shows that SPEA2 had the best average performance in terms of the $\varepsilon$-indicator and hypervolume unary metrics, the Mann-Whitney test showed that SPEA2’s results were not statistically different from $\varepsilon$-NSGAII. Relative to MOSCEM-UA, SPEA2 and $\varepsilon$-NSGAII attained superior results that were confirmed to be statistically different at the 99% confidence level.

Figures 3.11 and 3.12 show the dynamic results for the full distribution of trials and for the best single runs for the three algorithms, respectively. Performance metric differences between SPEA2 and $\varepsilon$-NSGAII resulted from a single trial run. As shown in Table 3.7 a single $\varepsilon$-NSGAII run failed to attain a top binary ranking, which is reflected in the upper bound of the shaded region in Figure 3.11. This single run highly biased both the mean and standard deviations for the unary
Figure 3.10. Hydrograph ranges corresponding to reference set of the Shale Hills test Case. The shaded area between dashed lines is the range of simulated hydrographs associated with the parameters groups in reference set. The bold line indicates the observed streamflow.

Table 3.7. Shale Hills case study’s ratios of top trial runs for each configuration of the algorithms based on the binary $\varepsilon$-indicator metric ranking. The best performing algorithm is highlighted in bold.

<table>
<thead>
<tr>
<th>MOEA Configurations</th>
<th>Top Ranking Ratios</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$-NSGAII (N=10)</td>
<td>14/15</td>
</tr>
<tr>
<td>SPEA2 (N=100)</td>
<td>15/15</td>
</tr>
<tr>
<td>MOSCEM-UA (N=250,C=2)</td>
<td>4/15</td>
</tr>
<tr>
<td>(N=250,C=5)</td>
<td>6/15</td>
</tr>
</tbody>
</table>

Table 3.8. Shale Hills case study’s results for the averages and standard deviations of the unary metrics for each algorithm configuration. AVG stands for mean, STD stands for standard deviation, and bolded entries highlight the best value attained.

<table>
<thead>
<tr>
<th>MOEA</th>
<th>Hypervolume</th>
<th>$\varepsilon$-Indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AVG</td>
<td>STD</td>
</tr>
<tr>
<td>$\varepsilon$-NSGAII</td>
<td>2.09E+04</td>
<td>1.82E+04</td>
</tr>
<tr>
<td>SPEA2</td>
<td><strong>1.63E+04</strong></td>
<td>7.17E+03</td>
</tr>
<tr>
<td>MOSCEM-UA</td>
<td>4.71E+04</td>
<td>1.93E+04</td>
</tr>
</tbody>
</table>
Figure 3.11. Shale Hills test case dynamic performance results for the unary $\varepsilon$-indicator distance metric versus total design evaluations. Mean performance is indicated by a solid line, the standard deviation by a dashed line, and the range of performance by the shaded region. The plots were generated using 15 trial runs for each algorithm.

Figure 3.12. Dynamic performance plots showing the best performing Shale Hills trial runs for each algorithm.

metrics given in Table 3.8 for $\varepsilon$-NSGAII. The Mann-Whitney test validates that the remaining $\varepsilon$-NSGAII trial runs were not statistically different from SPEA2. For MOSCEM-UA, Table 3.7 in combination with Figure 3.11 show that more than 60 percent of the algorithm’s trial runs failed to solve this test case. Figures 3.9 and 3.12 show that $\varepsilon$-NSGAII’s best runs were superior relative to the other algorithms’ results, generating nearly all of the reference set.

As was noted for the Leaf River case study, SPEA2’s performance for the Shale Hills test case is heavily impacted by its archive size. It has been widely recognized [35] that SPEA2’s k-means clustering diversity operator allows the algorithm to at-
tain highly diverse solution sets for high-order Pareto optimization problems (i.e., problems with 3 or more objectives). This operator is only active in the search process if the archive is sized appropriately, which in typical applications will require trial-and-error analysis. For this test case every trial run would require a week of computing time. It should be noted that $\varepsilon$-NSGAII automatically generates its archive size based on users’ precision goals for each objective. Additionally, the algorithm starts with a very small population size, which is automatically adjusted to enhance search effectiveness. The results presented in this study are conservative tests for the $\varepsilon$-NSGAII because SPEA2 and MOSCEM-UA initiate search with at least an order of magnitude advantage in search population.

### 3.6 Discussion

#### 3.6.1 Relative benefits and limitations of SPEA2

SPEA2 is an excellent benchmark algorithm for multiobjective hydrologic model calibration. Overall SPEA2 attained competitive to superior results for most of the problems tested in this study. The algorithm’s poorest performance occurred on the $T_4$ test function, which represents a severely difficult multimodal problem with 219 local fronts. SPEA2’s best overall performance occurred for the Leaf River case study where the algorithm was far more reliable relative to both the $\varepsilon$-NSGAII and MOSCEM-UA. The Leaf River test case is challenging because of a false front with a very high solution density. Our analysis showed that carefully setting the archive size for SPEA2 for this case study enabled the algorithm to fully exploit its k-means clustering diversity operator to spread solutions across the search space and more reliably escape the false nondominated front. For the Shale Hills test case, SPEA2 and $\varepsilon$-NSGAII had statistically equivalent performance metrics, although SPEA2 was slightly more reliable. SPEA2 is generally superior in performance relative to MOSCEM-UA.

The primary strengths of the SPEA2 algorithm lie in the algorithm’s search reliability and its diversity preservation operator as has been recognized in other studies. In this study, SPEA2 showed a limited sensitivity to its population sizing and search parameters. Other studies \[35, 40, 99\] have shown that SPEA2’s sensi-
tivity to population size often manifests itself in terms of a performance threshold for very difficult problems where the algorithm fails until the population is made sufficiently large. In this study, SPEA2’s poor performance on test function $T_4$ provides an example of this performance threshold. In these cases, it is very difficult to predict how to appropriately size SPEA2’s population. Significant trial-and-error analysis is required. The biggest challenge in maximizing the performance of SPEA2 lies in specifying an effective archive size without a priori knowledge of the Pareto set. In practice, this would require significant trial-and-error analysis, which is problematic for more complex, computationally intensive calibration applications.

### 3.6.2 Relative benefits and limitations of MOSCEM-UA

MOSCEM-UA was the least competitive of the three algorithms tested in this study failing to effectively solve either the standardized test function suite or the Shale Hills test case. MOSCEM-UA attained its best performance on the Leaf River case study, which was used in its development [53]. On the Leaf River case study, MOSCEM-UA was inferior to SPEA2 and statistically similar to $\varepsilon$-NSGAII. MOSCEM-UA did not contribute to any of the reference sets (i.e., the best overall solutions) for the two hydrologic calibration applications. The algorithm’s Markov Chain Monte Carlo sampler in combination with its shuffle complex search operator does not scale well for problems of increasing size and/or difficulty. MOSCEM-UA’s binary $\varepsilon$-indicator rankings for all three test cases show that the algorithm is not reliable even with significant increases in population size and complex number.

MOSCEM-UA’s primary strength is its estimation of the posterior parameter distributions for hydrologic model parameters (assuming the initial Gaussian assumptions made for hydrologic parameters are acceptable to users). Additionally, the algorithm has a limited number of parameters that need to be specified (i.e., the population size and number of complexes). MOSCEM-UA is however, critically sensitive to these parameters. The matrix inversion used in the algorithm’s stochastic search operators causes MOSCEM-UA’s efficiency to dramatically reduce with increases in population size and complexes. The algorithm is best suited for hydrologic model calibration applications that have small parameter sets and
small model evaluation times. In general, it would be expected that MOSCEM-UA’s performance would be met or exceeded by either SPEA2 or \( \varepsilon \)-NSGAII.

3.6.3 Relative benefits and limitations of \( \varepsilon \)-NSGAII

\( \varepsilon \)-NSGAII attained competitive to superior performance results relative to SPEA2 on the test function suite and the Shale Hills test case. Overall, \( \varepsilon \)-NSGAII generated the majority the reference sets (i.e., best overall solutions) for both hydrologic model calibration case studies. \( \varepsilon \)-NSGAII also had the best single run results for both of the calibration case studies as illustrated in Figures 3.7 and 3.12. The algorithm’s poorest performance occurred on Leaf River case study, in which its average performance was inferior to SPEA2 and statistically equivalent to MOSCEM-UA.

Although \( \varepsilon \)-NSGAII generated 58% of the reference set for the Leaf River test, its binary \( \varepsilon \)-indicator metric rankings (see Table 3.5) show that the algorithm performed less reliably than SPEA2. This highlights the biggest limitation impacting \( \varepsilon \)-NSGAII’s performance, which is related to its parent algorithm NSGAII’s diversity operator [101]. It has been widely reported [35, 40] that the original NSGAII converges very quickly, but its crowded tournament diversity operator can fail to promote sufficient diversity for some problems. Although Kollat and Reed [98] have demonstrated \( \varepsilon \)-NSGAII is statistically superior to the original NSGAII in terms of both convergence and diversity, \( \varepsilon \)-NSGAII can still be impacted by the limitations associated with the crowded tournament operator. For the Leaf River case study, \( \varepsilon \)-NSGAII had a reduced reliability relative to SPEA2 because several trial runs failed to create sufficiently diverse solutions that could escape the false local front. As was discussed above, SPEA2’s archive was sized carefully to maximize the effectiveness of its k-means clustering diversity operator, which allowed the algorithm to escape the local front. It is interesting to note that for the multimodal T4 test function with 219 local fronts, that \( \varepsilon \)-NSGAII’s performance is far superior to SPEA2. In this instance, \( \varepsilon \)-NSGAII’s was able to escape local fronts because of the random solutions injected into the search population during the algorithm’s dynamic changes in population size. In the limit, when the algorithm’s \( \varepsilon \)-dominance archive size stabilizes, the \( \varepsilon \)-NSGAII’s dynamic population sizing and random solution injection is equivalent to a diversity enhancing search operator.
termed “time continuation” [105].

In this study, ε-NSGAII appears to be superior to MOSCEM-UA and competitive with SPEA2 for hydrologic model calibration. ε-NSGAII’s primary strength lies in its ease-of-use due to its dynamic population sizing and archiving which lead to rapid convergence to very high quality solutions. Overall ε-NSGAII found a majority of the best known solutions for the calibration problems using less than 5,000 model evaluations. ε-NSGAII’s dynamic population sizing and archive-based preconditioning of search helps eliminate the need for trial-and-error analysis relative to SPEA2, which is particularly important for computationally intensive applications like the Shale Hills test case.

3.7 Conclusions

This chapter provides a comprehensive assessment of state-of-the-art evolutionary multiobjective optimization tools’ relative effectiveness in calibrating hydrologic models. Three test cases were used to compare the algorithms’ performances. The first test case was composed of a standardized suite of computer science test problems, which were used to validate the algorithms’ abilities to perform global search effectively, efficiently, and reliably for a broad range of problem types. The ε-NSGAII attained the best overall performance for the test function suite followed by SPEA2. MOSCEM-UA was not able to solve the test function suite reliably. The second test case is a benchmark hydrologic calibration problem in which the Sacramento soil moisture accounting model is calibrated for the Leaf River watershed. SPEA2 attained statistically superior performance for this case study in all metrics at the 99% confidence level. MOSCEM-UA and ε-NSGAII attained results that were competitive with another for the Leaf River case study. The third test case assessed the algorithms’ performances for a computationally intensive integrated hydrologic model calibration application for the Shale Hills watershed located in the Susquehanna River Basin in north central Pennsylvania. For the Shale Hills test case, SPEA2 and ε-NSGAII had statistically equivalent performance metrics, although SPEA2 was slightly more reliable. MOSCEM-UA’s performance on the Shale Hills test case was limited by the severe computational costs associated with increasing the algorithm’s population size and number of
complexes.

Overall, SPEA2 is an excellent benchmark algorithm for multiobjective hydrologic model calibration. SPEA2 attained competitive to superior results for most of the problems tested in this study. $\varepsilon$-NSGAII appears to be superior to MOSCEM-UA and competitive with SPEA2 for hydrologic model calibration. The largest challenge in using SPEA2 is effectively sizing the algorithm’s archive without prior knowledge of the true solution set. Trial-and-error analysis is required for maximizing the SPEA2’s performance. $\varepsilon$-NSGAII’s primary strength lies in its ease-of-use due to its dynamic population sizing and archiving which lead to rapid convergence to very high quality solutions with minimal user input. MOSCEM-UA is best suited for hydrologic model calibration applications that have small parameter sets and small model evaluation times. In general, it would be expected that MOSCEM-UA’s performance would be met or exceeded by either SPEA2 or $\varepsilon$-NSGAII.
Chapter 4

Study 2: Overcoming performance limits for multiobjective solution tools using parallelization strategies

This chapter is drawn from Y. Tang et al.’s paper which is in press in Advances in Water Resources [125]. The paper uses a formal metrics-based framework to demonstrate the Master-Slave (MS) and the Multiple-Population (MP) parallelization schemes for the Epsilon-Nondominated Sorted Genetic Algorithm-II (ε-NSGAII). It uses three test cases to compare the MS and MP versions of the ε-NSGAII: (1) an extremely difficult benchmark test function termed DTLZ6 drawn from the computer science literature, (2) an unconstrained, continuous hydrologic model calibration test case for the Leaf River near Collins, Mississippi, and (3) a discrete, constrained four-objective long-term groundwater monitoring (LTM) application.

4.1 Introduction

This chapter contributes a formal metrics-based framework to demonstrate the MS and the MP parallelization schemes for the ε-NSGAII. The ε-NSGAII has been demonstrated to be competitive to superior to other state-of-the-art evolutionary multiobjective (EMO) algorithms on a suite of water resources applications (see
The MS and MP versions of the \( \varepsilon \)-NSGAII developed in this chapter generalize the algorithm’s auto-adaptive population sizing, \( \varepsilon \)-dominance archiving, and time continuation to a distributed processor environment using the Message Passing Interface (MPI) parallelization libraries [65]. This chapter assumes that readers will have an introductory understanding of EMO algorithms (for an introduction see Chapter 2 of this thesis). A key finding of this work is that time-continuation and parallel speedups can dramatically improve the efficiency and reliability of EMO algorithms in water resources applications. Time continuation is an evolutionary algorithm (EA) search enhancement that promotes solution diversity and allows the \( \varepsilon \)-NSGAII to maintain effective search for as long as is necessary or is computationally tractable (more details are presented in Section 4.2.2). This chapter uses three test cases to compare the MS and MP versions of the \( \varepsilon \)-NSGAII: (1) an extremely difficult benchmark test function termed DTLZ6 drawn from the computer science literature [2], (2) an unconstrained, continuous hydrologic model calibration test case for the Leaf River near Collins, Mississippi [14, 17, 18, 26, 53, 97, 102], and (3) a discrete, constrained four-objective long-term groundwater monitoring (LTM) application [1, 98, 126]. These test cases were carefully selected to encompass a broad range of problem properties and clearly demonstrate the importance of problem difficulty when selecting parallelization strategies for multiobjective applications. Specifically, this chapter highlights that artificially constructed, extremely difficult test problems such as DTLZ6 may wrongly bias water resources professionals towards using more complicated algorithms (i.e., the MP parallelization scheme), when a simpler MS strategy may work as well or better for real-world problems.

There has been a modern confluence of systems analysis research towards approaches that emphasize multiple objectives (see reviews [35, 127–129]). This trend is clearly evident in the water resources literature over the past decade [39, 46, 52, 130]. Recent applications demonstrate that a growing body of researchers in both the water resources and the broader systems analysis communities are seeking to use EMO algorithms to solve large (in terms of the number of decisions and objectives), computationally intensive optimization problems [35, 39, 42, 131, 132]. Kollat and Reed [1] have recently shown that EMO algorithms potentially have a quadratic computational complexity when solving water resources applications.
A quadratic complexity implies that a $k$-fold increase in the number of decision variables will yield an $k^2$-fold increase in the number of function evaluations (NFE) required to solve an application. Kollat and Reed’s findings motivate the need to effectively design and comprehensively assess EMO parallelization schemes that can help to overcome the computational constraints posed by large water resources problems. There is a dearth of parallel EMO studies in the water resources literature despite the large number of areas where multiobjective applications are prevalent. The Leaf River hydrologic calibration test case and the LTM test case were selected in this chapter to represent two distinctly different problem areas within the water resources literature where there is substantial interest in advancing multiobjective methods.

The Leaf River test case is a benchmark hydrologic model calibration problem in which the Sacramento soil moisture accounting model (SAC-SMA) is calibrated for the watershed located close to Collins, Mississippi. The Leaf River case study has been used in the development of both single and multiobjective calibration tools. The growing body of research in the area of multiobjective calibration has shown that the multiobjective approach is practical, relatively simple to implement, and can provide insights into parameter uncertainty as well as the limitations of a model. Although a majority of prior studies have focused on conceptual rainfall-runoff applications, there have been an increasing number of recent studies focused on developing multiobjective calibration strategies for computationally intensive distributed hydrologic models where effective EMO parallelization strategies are needed. Readers should also note that our Leaf River analysis is conservative in the sense that the SAC-SMA evaluation times are relatively small, making it more difficult to attain efficient parallel speedups.

Groundwater monitoring design was one of the first EMO application areas in the water resources literature. In general, groundwater monitoring design has been shown to be a challenging optimization problem with several conflicting objectives and very large discrete decision spaces. Knopman and Voss recognized that the groundwater quality network design problem has many mathematical similarities to the classical combinatorial knapsack problem (i.e., discrete decision spaces that grow exponentially with increased problem...
size). Reed and Minsker [39] used the LTM problem to illustrate that EMO algorithms are capable of solving a new problem class [131, 132], which they termed high-order Pareto optimization. The term high-order Pareto optimization is used to describe those applications that seek to quantify optimal tradeoffs between three or more objectives. This paper uses the LTM test case to demonstrate that the parallel versions of the \( \varepsilon \)-NSGAII can rapidly and reliably solve high-order Pareto optimization problems. More generally, the three test cases used in this chapter demonstrate that the MS and MP versions of the \( \varepsilon \)-NSGAII can broaden the size and difficulty of multiobjective water resources applications that can be solved efficiently and reliably.

This chapter proceeds as follows. The multiobjective optimization methods, parallelization strategies, test cases, performance metrics, and computational experiment are discussed in Section 4.2. Section 4.3 presents the results of the study followed by a discussion in Section 4.4 comparing the MS and MP versions of the \( \varepsilon \)-NSGAII. Finally, the conclusions of this chapter are presented in Section 4.5.

4.2 Methodology

4.2.1 Evolutionary multi-objective optimization search

Evolutionary multiobjective optimization algorithms are similar to traditional single objective evolutionary algorithms in that all genetic algorithms search complex problem spaces using a process that is analogous to Darwinian natural selection. Evolutionary algorithms use a population-based search in which high quality solutions are evolved using the three basic operators of (1) selection, (2) mating, and (3) mutation. Analogous to natural systems, selection preferentially samples higher fitness solutions and biases the population to converge to the best solutions. The fitness of each solution is determined by how well it satisfies specified objectives and constraints of a given application. Mating occurs by combining the decision variables of high quality “parent” solutions to create “child” solutions. The mating operator in combination with selection allows evolutionary algorithms (EAs) to globally search promising regions of a problem space. Lastly, mutation perturbs the decision variables that compose population members. Selection in combina-
tion with mutation allows EAs to locally search the problem space near a given solution. The primary difference between EMO algorithms and single objective EAs lies in how fitness is assigned.

EMO algorithms assign solutions fitness values based on their performance across a vector of objectives. A solution cannot be assessed in terms of its performance in any single objective because it may perform poorly with respect to the remaining objectives. Instead, the concept of Pareto dominance is used to assign fitness values to solutions. A solution \( x \) dominates another solution \( x' \) if and only if it performs as well as \( x' \) in all objectives and better in at least one. Solutions are assigned ranks based on their Pareto dominance where top performing solutions are non-dominated (i.e., no solution exceeds their performance in all objectives). After Pareto ranking, EMO algorithms apply selection to bias search towards top ranking non-dominated solutions. In cases where solutions have equal ranking, selection biases search towards “diverse” solutions that are distant from neighboring solutions. Solution diversity is a key factor in finding solutions along the full extent of an application’s tradeoffs.

More formally, the goal of multiobjective optimization is to identify the Pareto-optimal tradeoffs between an application’s objectives. These tradeoffs are composed of the set of solutions that are better than all other solutions in at least one objective and are termed non-dominated or Pareto-optimal solutions \([147]\). The Pareto-optimal front is obtained by plotting these solutions according to their objective values yielding an \( M - 1 \) dimensional surface where \( M \) is the total number of objectives. The term high-order Pareto surfaces is used to describe those surfaces that result from three or more objectives. EMO algorithms’ population-based search enables them to evolve entire tradeoff (or Pareto) surfaces within a single run for problems with huge decision spaces. These methods can solve highly nonlinear, discrete, and non-convex problems without differentiation \([36–38]\).

### 4.2.2 The \( \varepsilon \)-NSGAII

The \( \varepsilon \)-NSGAII has been demonstrated in both discrete and continuous water resources applications \([97, 98]\). The \( \varepsilon \)-NSGAII’s performance has been shown to be superior to the original Non-Dominated Sorted Genetic Algorithm II (NS-
GAII) [101] and the Epsilon-Dominance Multi-Objective Evolutionary Algorithm ($\varepsilon$MOEA) [148] and competitive to superior relative to the Strength Pareto Evolutionary Algorithm 2 (SPEA2) [99] on a suite of water resources applications. All of these MOEAs use real parameter simulated binary crossover (SBX), polynomial mutation, and elitism [41, 149]. The primary goal in the development of the $\varepsilon$-NSGAII was to provide a highly reliable and efficient EMO algorithm which minimizes the need for traditional EA parameterization and allows the user to focus on problem specific search quality goals. Computational savings can be viewed in two contexts: (i) the use of minimal population sizes and (ii) the elimination of trial-and-error application runs to determine search parameters.

The $\varepsilon$-NSGAII builds on its parent algorithm, the NSGAII [101], by adding $\varepsilon$-dominance archiving [103, 148] and adaptive population sizing [104] to minimize the need for extensive parameter calibration as demonstrated by Reed et al. [104]. The concept of $\varepsilon$-dominance allows the user to specify the precision with which they want to quantify each objective in a multiobjective problem. Figure 4.1 demonstrates the concept of $\varepsilon$-dominance using a two step approach. First, a user specified $\varepsilon$ grid is applied to the search space of the problem. Larger $\varepsilon$ values result in a coarser grid (and ultimately fewer solutions) while smaller $\varepsilon$ values produce a finer grid. Grid blocks containing multiple solutions are then examined and only the solution closest to the bottom left corner of the block is kept (assuming minimization of all objectives). In the second step, non-domination sorting based on the grid blocks is then conducted resulting in a “thinning” of solutions and
promoting a more even search of the objective space. Epsilon-dominance allows the user to define objective precision requirements that make sense for their particular application. The interested reader can refer to prior work by Laumanns et al. [103] and Deb et al. [148] for a more detailed description of $\varepsilon$-dominance.

The $\varepsilon$-NSGAII uses a series of “connected runs” where small populations are initially exploited to pre-condition search and automatically adapt population size commensurate with problem difficulty. As the search progresses, the population size is automatically adapted based on the number of $\varepsilon$-nondominated solutions that the algorithm has found. Epsilon-non-dominated solutions found after each generation are stored in an archive and subsequently used to direct the search. Although the adaptation of population size will differ depending on the random seed chosen, exploiting small populations to precondition search will on average greatly reduce computation times. Theoretically, this approach allows the population size to increase or decrease, and in the limit when the $\varepsilon$-dominance archive size stabilizes, the $\varepsilon$-NSGAII’s “connected runs” are equivalent to a diversity-based EA search enhancement recommended by Goldberg [38] termed “time continuation”. In this study, search was terminated across all runs (i.e., across all populations used) after a user-specified maximum run time has been reached.

The $\varepsilon$-NSGAII uses time continuation to enhance population diversity and extend the time of active exploration. Time continuation results from combining random solutions with $\varepsilon$-dominance archive members each time the population size is adapted. Each new population is equal to four times the current $\varepsilon$-dominance archive’s size so that $\varepsilon$-nondominated archive members compose 25-percent of the new population and the remaining 75-percent is composed of randomly generated solutions. This aspect of the $\varepsilon$-NSGAII’s search is particularly important in the context of computational demands and parallelization. Time continuation allows the algorithm to maintain effective search for as long as is necessary or computationally tractable. For many water resources applications, problem difficulty and/or the computational expense of solution evaluations heavily impact search time constraints and solution quality. This chapter specifically demonstrates parallelization schemes for the $\varepsilon$-NSGAII that effectively exploit the algorithm’s $\varepsilon$-dominance archiving, adaptive population sizing, solution injection, and time continuation to dramatically improve solution quality, search efficiency, and algo-
rithmic reliability.

4.2.3 Parallelization strategies for the $\varepsilon$-NSGAII

Parallel computing is well suited for overcoming the computational bottleneck posed by computationally expensive objective functions while also improving the effectiveness and reliability of EMO algorithms \cite{35, 68, 70}. There are three main parallel paradigms for EMO algorithms \cite{35, 68}: the master-slave model, the multi-population model, and the diffusion model. These models can be hybridized and their implementations vary significantly in the computer science literature. In this chapter, the MS and MP models are adopted since they are better suited for smaller scale clusters ($<$ 50 processors). In this study, we used a Linux computing cluster with 133 computer nodes composed of dual or quad AMD Opteron processors and 64 GB of RAM. The cluster runs the GNU/Linux operating system and is installed with the Argonne National Laboratory’s Infinicon Message Passing Interface (MPI). The Infinicon MPI runs natively over a very high-bandwidth, ultra low-latency network interconnect. This application utilized up to 16 processors to represent the typical size of computer clusters.

4.2.3.1 The Master-Slave $\varepsilon$-NSGAII

The MS parallelization scheme [see Figure 4.2] is the simplest potential parallelization strategy for EAs. In this scheme, the master processor has a fully functional version of the $\varepsilon$-NSGAII that uses slave processors to evaluate solutions and return objective values. The master processor uses the objective function values to perform all of the required evolutionary search operations (ranking, selection, mating, and mutation). In our master-slave version of the $\varepsilon$-NSGAII, the master processor also evaluates potential solutions while managing issues associated with slave synchronization and load balance (i.e., all slaves have approximately equal workload). When the master reaches the point of fitness function evaluation, it tells all of the processors to be ready to receive the individuals for evaluation and “passes a message” containing the variables of potential solutions to the slaves. After evaluating its own group of individuals, the slave sends the objective values back to the master and the master uses all of the evaluated objectives’ values to
continue the evolutionary process until a stop criterion is met. The MS paradigm has the advantage of being extremely easy to implement. Prior criticisms of the MS model in the EA literature (for reviews see [72, 150]) include:

- Communication costs increase significantly with an increasing number of processors.

- The parallelization scheme does not change the serial algorithm’s expected performance.

- In multiobjective applications, the slowest slave processor will control the overall efficiency and timing of the remaining processors because nondomination sorting is delayed until the slowest slave processor returns its evaluation.

Water resources researchers should carefully consider if these criticisms generalize to their own applications. For example, while it is generally true that the master-slave parallelization scheme will result in the same search dynamics as the serial algorithm, it should be noted that the time savings associated with parallelization can allow for significantly more search. If an application requires four hours to complete 100,000 evaluations, then four processors could potentially evaluate 400,000 solutions in the same four hour period. This is particularly important for the $\varepsilon$-NSGAII where time-continuation can serve to maintain sufficient diversity to support more extensive search for as long as is necessary or feasible when solving water resources applications. Synchronization problems (i.e., search is delayed until the slowest slave processor send its evaluation) occur for MS applications when the average solution evaluation time $T_f$, has a high variance. Communication costs are of primary concern when judging the efficacy of a parallel application using
the concept of speedup, \( S_p \), defined in Eq. 4.1

\[
S_p = \frac{T_s}{T_p}
\]

(Eq. 4.1)

Speedup is used to judge parallel performance by comparing the clock time required to solve an application in serial (i.e., on one processor), \( T_s \), with the clock time required using multiple processors, \( T_p \). As highlighted by Cantu-Paz [72], it is very important when judging speedup that solution quality should also be monitored. Monitoring solution quality will ensure that prematurely converged results with small clock times and poor solution quality do not bias speedup assessments. Ideally, the goal of parallelization is to attain “linear speedups” which means that when \( P \) processors are used to solve an application, the parallel computing time \( T_p \), will equal \( \frac{1}{P} \times T_s \) (i.e., speedup is equal to the number of processors used). In each generation of MS search, Eq. 4.2 from Cantu-Paz [72] shows that the parallel processing time \( T_p \), and speedups are a function of the population size \( n \), the average solution evaluation time \( T_f \), the number of processors \( P \), and the communication time \( T_c \) associated with passing messages.

\[
T_p = PT_c + \frac{nT_f}{P}
\]

(Eq. 4.2)

Eqs. 4.1 and 4.2 show that as the number of processors increase, there exists an asymptotic limit to the speed-ups that can be attained with the MS parallelization scheme. It should be noted that many water resources applications have solution evaluation times that are significantly larger than communication times (i.e., \( T_c << T_f \)) yielding very good speed-ups for master-slave EMO applications.

4.2.3.2 The Multi-Population \( \varepsilon \)-NSGAII

The MP parallelization scheme is often called the island model, which is a reference to the biological theory of punctuated equilibrium [151, 152] formulated to model rapid evolutionary changes that result from isolated populations. The approach utilizes multiple populations distributed on different processors, each of which has a fully functional version of the \( \varepsilon \)-NSGAII. Recent literature reviews [35, 72] highlight that multiple population schemes have emerged as one of the most
popular parallelization strategies for evolutionary algorithms due to the emerging prevalence of distributed computing clusters and because early studies showed the potential for “superlinear” speed-ups (e.g., an algorithm is five times faster using only four processors). Some basic terms used in the multi-population EA literature are provided below:

- **Deme** refers to the population assigned to each processor.

- **Epoch** refers to the number of generations a deme evolves without communicating with other demes.

- **Topology** describes “processor connectivity” or the rules for communication between demes.

- **Migration rate** specifies how many solutions a deme will share with other demes.

- **Migration frequency** specifies how often migration will occur.

The terms defined above highlight that standard MP implementations of both single and multiobjective EAs significantly increase the design and parameterization issues that users must address to attain rapid and reliable search. The auto-adaptive population sizing, \( \varepsilon \)-dominance archiving, and solution injection methods used in the serial \( \varepsilon \)-NSGAII were generalized to a \( P \)-processor environment, simplifying the design and parameterization of the MP version of the algorithm. As illustrated in Figure 4.3, the major difference between the MP version and the
serial version of the $\varepsilon$-NSGAII is that the parallel algorithm implements adaptive population sizing, dynamic archiving, and solution injection across multiple processors. Worker processors and the coordinator evolve populations based on different random initializations. Each processor searches the entire parameter space for the Pareto front (i.e., every processor has its own fully operational version of the $\varepsilon$-NSGAII). Please note that the MP version of the $\varepsilon$-NSGAII automatically and dynamically adapts deme sizes, epochs, communication topologies, migration rates, and migration frequencies without additional user-inputs.

The MP version of $\varepsilon$-NSGAII uses asynchronous and dynamic messaging to automatically migrate solutions between processors and adjust deme sizes (see Fig. 4.3). Each processor starts with an arbitrarily small deme size (i.e., 12 members) and initiates search for 250 generations (i.e., the maximum epoch). After the first full epoch, the worker processors can then send requests to the coordinator for global archive solutions (initially representing a fully connected topology and migration strategy). Next, the coordinator collects the solutions from all of the local workers’ archives to update the global archive using $\varepsilon$-non-domination sorting. The coordinator then sends the global archive solutions back to the processors who requested population changes. The new population sizes for each worker are four times the updated global archive size. One fourth of the individuals are injected from the global archive into each new population, and the remaining individuals are generated randomly. After the first 250 generation epoch, the worker processors can make asynchronous and dynamic population sizing requests if within ten generations they fail to improve more than 10-percent of their local $\varepsilon$-dominance archived solutions. When search progress is insufficient, each worker will send a request to the coordinator and will receive a new population of which 25-percent of the solutions are global archive members and the remaining 75-percent are randomly generated. Migration rate and frequency are determined dynamically based on each deme’s search progress as measured by how quickly it updates its local archive. Overall, search termination is the responsibility of the coordinator processor once the termination criterion is satisfied (i.e., maximum clock time). The coordinator sends out termination messages to all of the workers if the criterion is met. The communications of global archive solutions or termination signals are conducted by using token loops [153] to avoid processor deadlocks.
Attaining efficient speedups for a multiple population EA is challenging and related directly to computing hardware, decisions on parameter settings, and the difficulty of the application being solved. Eq. 4.3 provides a simplified but informative description of the factors impacting parallel computation time $T_p$, for MP parallelization schemes.

$$T_p = gn_d T_f + (r - 1) T_c$$

The total parallel computation time for a multiple population EA is the sum of the time required for each deme to evolve ($gn_d T_f$) and the time spent communicating between processors ($(r - 1) T_c$). The component of parallel computation time spent evolving demes is given by the product of the deme size $n_d$, the number of generations $g$, and the average time required per evaluation $T_f$. The second component of Eq. 4.3 approximates the total time spent sharing or communicating solutions among $r - 1$ demes. $T_c$ is the average clock time required per communication. Eq. 4.3 highlights key design and parameterization issues that impact a multiple-population EA’s ability to attain efficient speedups. Minimizing the deme size ($n_d$) will help to reduce parallel computation times and enhance speedups. Unfortunately, a small deme or population size will often cause EAs to have a low reliability [73]. Substitution of Eq. 4.3 into Eq. 4.1 also shows that as communication between processors increases, there will be a degradation in parallel speedup. This degradation causes parallel performance to again be “asymptotic” as the number of processors increase. In other words, increasing communication costs by using more processors will result in an upper bound limit where adding more processors will not improve $T_p$. The speedup asymptote is largely controlled by the ratio of function evaluation time and communication time. It is easier to attain efficient speedups when this ratio is large (i.e., $T_f \gg T_c$).

### 4.2.4 Case studies

The test cases used in this chapter were selected to encompass a broad range of problem properties. The DTLZ6 test function [2] is a benchmark three-objective problem in the EMO literature with a known Pareto front. DTLZ6 is one of the most difficult problems in the DTLZ suite due to its deceptive and multi-
modal search space (i.e., false fronts that cause premature convergence). Prior literature \cite{2, 148} has shown that modern EMO algorithms are unreliable in exactly quantifying DTLZ6’s global Pareto surface. The second case study is a benchmark hydrologic model calibration problem based on long-term data for the Leaf River in Collins, Mississippi. The Leaf River test case has been used extensively in the water resources literature to develop and test optimization algorithms \cite{14, 17, 18, 53, 97, 102}. More recently, Tang et al. \cite{97} have shown that the Leaf River case study is difficult for modern serial EMO algorithms, including the \(\varepsilon\)-NSGAII, to solve reliably. The Leaf River case study represents an unconstrained, continuous problem that is both deceptive and multimodal (see \cite{97}). The final test case, a long-term groundwater monitoring (LTM) application represents a challenging discrete and constrained problem that has been used extensively in the development and testing of the serial version of the \(\varepsilon\)-NSGAII \cite{98, 138}. Nearly half of this problem’s search space is infeasible and it has a very large four-objective Pareto optimal surface that has been identified through enumeration. Each of these case studies and their problem formulations are presented in detail in sections 4.2.4.1 through 4.2.4.3.

4.2.4.1 Case 1: Test problem DTLZ6

The three-objective formulation of DTLZ6 is shown in Eq. 4.4 where the \(x_i\) and \(\theta_i\) are decision variables and \(f_i\) is the objective function.

\[
\begin{align*}
\text{Min. } f_1(x) &= (1 + g(x_M)) \cos(\theta_1/2) \cos(\theta_2/2) \cdots \cos(\theta_{M-2}/2) \cos(\theta_{M-1}/2) \\
\text{Min. } f_2(x) &= (1 + g(x_M)) \cos(\theta_1/2) \cos(\theta_2/2) \cdots \cos(\theta_{M-2}/2) \sin(\theta_{M-1}/2) \\
\text{Min. } f_3(x) &= (1 + g(x_M)) \cos(\theta_1/2) \cos(\theta_2/2) \cdots \sin(\theta_{M-2}/2)
\end{align*}
\]

where \(\theta_i = \frac{\Pi}{4(1 + g(r))}(1 + 2g(r)x_i), \text{ for } i = 2, 3, \cdots, (M - 1)\) \hfill (4.4)

\[
g(x_M) = \sum_{x_i \in x_M} x_i^{0.1},
\]

\[0 \leq x_i \leq 1, \text{ for } i = 1, 2, \cdots, n\]

There are twelve decision variables as recommended by Deb \cite{2}. Although this is a three-objective problem, the true Pareto front of this test function is a curved
Figure 4.4. Pareto front of DTLZ6 and the non-dominated region of point A and B on the front. The shaded region is the non-dominated region of A and B. Adapted from [2].

line surrounded by larger false fronts. As illustrated in Figure 4.4, EMO algorithms tend to pre-converge to the large local non-dominated areas intersecting the Pareto front. For example, if only solutions A and B on the true Pareto front are found, then a very large non-dominated region exists (indicated by the shaded area). The true Pareto optimal solutions comprise a very small fraction of the overall number of nondominated solutions that exist between A and B yielding a high probability of failure for search algorithms.

4.2.4.2 Case 2: Model calibration in the Leaf River watershed

The Leaf River SAC-SMA test case represents a benchmark problem within the water resources literature that has been used extensively for developing tools and strategies for improving hydrologic model calibration [14, 17, 18, 26, 53, 97, 102]. Readers interested in the full details of the Leaf River case study’s dataset should reference earlier works (e.g., [112]). The Leaf River case study used in this chapter has been developed based on the original studies used to develop and demonstrate MOSCEM-UA [53, 102]. The Sacramento Soil Moisture Accounting model is a 16 parameter lumped conceptual watershed model used for operational river forecasting by the National Weather Service throughout the US (see [113], for more
details on the model). The algorithm searched the same 13 SAC-SMA parameters (three parameters are commonly fixed a priori) and parameter ranges as recommended by Vrugt et al. The algorithm is tested based on its ability to quantify a two-objective tradeoff using a root-mean square error (RMSE) problem formulation. The first objective was formulated using a Box-Cox transformation of the hydrograph as recommended by Misirli et al. to reduce the impacts of heteroscedasticity in the RMSE calculations (also increasing the influence of low flow periods). The second objective was the non-transformed RMSE objective, which is largely dominated by peak flow prediction errors due to the use of squared residuals. The objective functions are shown in Eq. 4.5.

\[
\begin{align*}
\text{Min. } f_1(\theta) &= \left[ \frac{1}{N} \sum_{i=1}^{N} (Q_{\text{obs},i} - Q_{\text{sim},i}(\theta))^2 \right]^{1/2} \\
\text{Min. } f_2(\theta) &= \left[ \frac{1}{N} \sum_{i=1}^{N} [T(Q_{\text{obs},i}) - T(Q_{\text{sim},i}(\theta))]^2 \right]^{1/2}
\end{align*}
\]

(4.5)

where \(Q_{\text{obs},i}\) is the observed discharge at time \(i\), \(Q_{\text{sim},i}(\theta)\) is the simulated discharge, and \(N\) is the total number of time steps in the calibration period. \(T\) is the Box-Cox transformation function \(T(f) = [(f + 1)^{\lambda} - 1]/\lambda\) where \(\lambda = 0.3\).

A 65-day warm-up period was used based on the methodological recommendations of Vrugt et al. A ten-year calibration period was used from 1 October 1952 to 30 September 1962. In the application, 50 trials were used for the serial run (one processor) and multi-processor runs of the MS and MP versions of the \(\varepsilon\)-NSGAII (2, 4, 8, 16-processors settings were tested), yielding a total of 450 EMO algorithm trial runs. Each EMO algorithm trial run utilized a maximum of \(P \times 100,000\) (\(P\) is the number of processors) SAC-SMA model evaluations, yielding a total of 305,000,000 SAC-SMA model evaluations used in our Leaf River case study analysis. A best known approximation set was generated for this problem by conducting non-dominated sorting on the final results collected from all of the trial runs. The best known solution set for the Leaf River case study is shown in Figure 4.5.
Figure 4.5. Reference set generated for the Leaf River test case where RMSE(T) are the errors for the Box-Cox transform of the hydrograph and RMSE(R) are the errors for the raw hydrograph.

4.2.4.3 Case 3: Long-term groundwater monitoring design

The LTM test case used in this chapter is based on a 50-million node flow and transport simulation originally developed by Maxwell et al. [154]. This test case represents the migration of a hypothetical perchloroethylene (PCE) plume originating from an underground storage tank. The hydrogeology of the site has been extensively characterized and is based on a highly heterogeneous alluvial aquifer located at the Lawrence Livermore National Laboratory in Livermore, California. Concentration data are provided at 58 hypothetical sampling locations in a 29 well monitoring network for a snapshot in time 8 years following the initial release of contaminant. Each well can be sampled from one to three times along its vertical axis and the sampling domain extends 650 m in the $x$ direction, 168 m in the $y$ direction, and 38.4 m in the $z$ direction with a minimum horizontal spacing of 10 m between wells. Additional details on this test case can be found in Reed et al. [126]. Four design objectives were chosen for this study, each of which were minimized: (i) sampling cost, (ii) relative error of local contaminant concentration estimates, (iii) local contaminant concentration estimation uncertainty, and (iv) contaminant mass estimation error. Objectives (ii)-(iv) were obtained using the Quantile Kriging method. Eq. (4.6) represents the objective formulation where $F(x_k)$ is a vector valued performance function in which the four objectives: cost
(\(f_{\text{cost}}\)), concentration estimation error (\(f_{\text{conc}}\)), local uncertainty (\(f_{\text{uncert}}\)), and mass estimation error (\(f_{\text{mass}}\)) are minimized. Eq. (4.7) subjects \(F(x_\kappa)\) to the constraint that no points in the interpolation domain remain unestimated (which may occur if a particular sampling plan contains too few sampling points to successfully Krige the entire domain).

Minimize

\[
F(x_\kappa) = \left( f_{\text{cost}}(x_\kappa), f_{\text{conc}}(x_\kappa), f_{\text{uncert}}(x_\kappa), f_{\text{mass}}(x_\kappa) \right), \quad \forall \kappa \in \Omega \quad (4.6)
\]

Subject to \(U(x_\kappa) = 0 \quad (4.7)\)

The objectives are all a function of the vector \(x_\kappa\) representing the \(\kappa\)th sampling plan in the decision space \(\Omega\). Each component \(i\) of a sampling plan \(\kappa\) is determined from Eq. (4.8) resulting in a string of binary digits indicating whether or not a well is sampled.

\[
x_{\kappa,i} = \begin{cases} 
1, & \text{if the ith well is sampled} \\
0, & \text{otherwise} 
\end{cases}, \quad \forall \kappa, i \quad (4.8)
\]

The sampling cost objective quantifies the monitoring cost of a particular sampling scheme using Eq. (4.9). The coefficient, \(C_S\) defines the cost per sample (normalized to one in this study). Additionally, if a well is sampled, it is assumed that all locations along its vertical axis are sampled resulting in a cost coefficient ranging from 1 to 3. The cost objective is ultimately quantified by summing the cost coefficients of each of the wells sampled in a particular scheme.

\[
f_{\text{cost}}(x_\kappa) = \sum_{i=1}^{\text{nwell}} C_S(i) x_{\kappa,i} \quad (4.9)
\]

The relative error of local contaminant concentration estimates objective measures how the Kriged estimate of the plume using the \(\kappa\)th sampling plan differs from that obtained by sampling from all well locations. Eq. (4.10) quantifies the concentration error objective by summing the squared differences between the concentration estimate at a grid location \(u_j\) using all wells, \(c_{\text{all}}(u_j)\), and the concentration esti-
mate at the same grid location using the $\kappa$th sampling plan, $c_\kappa(u_j)$.

$$f_{\text{conc}}(x_\kappa) = \sum_{j=1}^{n_{\text{est}}} (c_{\text{all}}(u_j) - c_\kappa(u_j))^2$$  \hspace{1cm} (4.10)

Local contaminant concentration estimation uncertainty is quantified by summing the estimation standard deviations obtained from Kriging at each grid location $u_j$ using Eq. (4.11). The standard error weight coefficient, $A_j$, can be used to assigned importance to uncertainty estimates at different locations in the interpolation domain. For this study, $A_j$ was assumed constant across the interpolation domain and was assigned a value of $2\sqrt{3}$ based on the standard deviation of a uniform distribution.

$$f_{\text{uncert}}(x_\kappa) = \sum_{j=1}^{n_{\text{est}}} A_j \sigma(u_j)$$  \hspace{1cm} (4.11)

The contaminant mass estimation error objective quantifies the relative error between the total mass of dissolved contaminant estimated using all well locations, $\text{Mass}_{\text{all}}$, and the contaminant mass estimated from the $\kappa$th sampling plan, $\text{Mass}_\kappa$. Eq. (4.12) expresses the relative mass estimation error in terms of a percentage.

$$f_{\text{mass}}(x_\kappa) = \left| \frac{\text{Mass}_{\text{all}} - \text{Mass}_\kappa}{\text{Mass}_{\text{all}}} \right| \cdot 100\%$$  \hspace{1cm} (4.12)

If a well sampling scheme results in unestimated points in the interpolation domain [violating the constraint described by Eq. (4.7)], the objectives are penalized to ensure that infeasible sampling schemes are eliminated from consideration. Eq. (4.13) is applied to each objective function if a feasibility violation occurs, resulting in solutions with lower fitness (i.e., higher objective values in a minimization problem).

$$F_{\text{penalty}}(x_\kappa) = \begin{cases} f_{\text{penalty}}^\text{cost} = f_{\text{cost}} + f_{\text{max}}^\text{cost} \\ f_{\text{penalty}}^\text{conc} = f_{\text{conc}} + n_{\text{est}} + U(x_\kappa) + f_{\text{max}}^\text{cost} \\ f_{\text{penalty}}^\text{uncert} = f_{\text{uncert}} + n_{\text{est}} + U(x_\kappa) + f_{\text{max}}^\text{cost} \\ f_{\text{penalty}}^\text{mass} = f_{\text{mass}} + n_{\text{est}} + U(x_\kappa) + f_{\text{max}}^\text{cost} \end{cases}$$  \hspace{1cm} (4.13)

Eq. (4.13) penalizes the objective functions based on the maximum cost of a sampling scheme, the total number of estimation points in the grid, and the total number of unestimated points, $U(x_\kappa)$, in the infeasible sampling plan. Penalizing
solutions rather than eliminating them ensures that sampling schemes which are “almost” feasible are given the opportunity to be further evolved by the MOEA into feasible designs (for more details on this problem formulation, see Reed and Minsker [39]).

Spatial interpolation of the contamination plume was conducted using Quantile Kriging (QK) based on the recommendations of Reed et al. [126]. Kriging provides a minimum error variance estimate value at an unsampled location provided the data at the sampled locations [155]. QK extends Ordinary Kriging (OK) by transforming the sample values to quantile space according to their rank. The quantile values represent the empirical cumulative distribution function (CDF) of the sample values, resulting in normalized data. Samples are Kriged in quantile space and then transformed back to concentration space using the generated CDF [156, 157]. Since OK assumes stationarity of the concentration mean, moving local neighborhoods are used to estimate the expected value at each location [155]. Reed et al. [126] found that QK showed the least bias with respect to variability of PCE concentrations and preferential sampling, and was most robust in representing the plume when compared to five other interpolation methods.

4.2.5 Performance metrics

When judging the performances of the parallel versions of the $\varepsilon$-NSGAII, it is important to monitor both speedup and solution quality. For EMO applications, solution quality metrics must consider both convergence and diversity. Convergence metrics quantify how distant an approximation set is from the reference set of optimal solutions and diversity metrics measure how well the approximation set’s solutions capture the full extent of the Pareto front. In this chapter, runtime convergence and diversity metrics [122] were used to measure these aspects of algorithm performance separately. The runtime $\varepsilon$-performance [100] and $\varepsilon$-indicator [108, 123] metrics were also used in this chapter to quantify these aspects of performance simultaneously.

The runtime convergence metric proposed by Deb and Jain [122] is used to quantify the average convergence of the algorithm to some reference set. This metric measures convergence using the average normalized Euclidean distance be-
tween the algorithms solutions and the optimal reference set of solutions. This metric was normalized to have a minimum value of zero for perfect convergence and a maximum value of one indicating very poor convergence. The diversity metric also proposed by Deb and Jain [122] measures how well the approximation set captures the full extent of the tradeoffs between a case study’s objectives. Deb and Jain’s diversity metric ranges from a maximum of one to a minimum of zero. A value of one represents a perfect solution diversity in terms of the metric formulation and user specified parameters. The metric is calculated by projecting the non-dominated solutions obtained during a single run of an EMO algorithm as well as the reference solutions onto a hyper-plane that is partitioned into grids. The metric is then computed by counting the number of non-dominated solutions which fall into the same grids as the solutions in the reference set. A detailed description of this metric can be found in [122].

The $\varepsilon$-performance metric proposed by Kollat and Reed [100] accounts for both convergence and diversity simultaneously. This metric ranges between zero and one where a metric value of one would indicate 100-percent convergence to within user specified $\varepsilon$ values of the reference set. Epsilon-performance counts the number of approximate set solutions that fall into $\varepsilon$ hyperboxes defined around the reference set and is calculated as the percentage of solutions which have successfully converged to the Pareto front within the user specified precision. For more details on this metric, see [98]. The $\varepsilon$-indicator metric [108, 123] represents the smallest distance that an approximation set must be translated to dominate the reference set, implying that smaller indicator values are preferred. Theoretically, the $\varepsilon$-indicator accounts for both convergence and the distribution of the solutions. For a more detailed description of this metric, see [108, 123].

4.2.6 Description of computational experiment

The $\varepsilon$-NSGAII used simulated binary crossover and polynomial mutation operators for all three test cases. Based on the most commonly recommended settings for these operators [2, 40, 101, 110, 122], the crossover probability was set equal to 1.0 and the probability of mutation was set equal to $1/L$ where $L$ is the number of decision variables. The distribution indices for crossover and mutation were set
equal to 15 and 20, respectively. An initial population size of 12 was used for the MS and the MP versions of the $\varepsilon$-NSGAII in all applications. Table 1 provides a detailed list of the parameters used for both parallel versions of the $\varepsilon$-NSGAII.

Epsilon resolution settings were chosen based on application specific resolution goals. For the DTLZ6 test case, uniform $\varepsilon$ values equal to 0.0045 were specified for all objectives so that the Pareto optimal set could be represented discretely by 100 solutions. For the Leaf River test case, the $\varepsilon$ values were set to $10^{-6}$ representing the minimum meaningful RMSE difference that should be resolved. In the LTM test case, $\varepsilon$ values were specified so that the $\varepsilon$-NSGAII could quantify the Pareto optimal set at the same resolution as was used in our enumeration. Each of the four objectives had the following $\varepsilon$ values $[\varepsilon_{\text{cost}} = 10^0, \varepsilon_{\text{conc}} = 10^{-5}, \varepsilon_{\text{uncert}} = 10^{-2}, \text{ and } \varepsilon_{\text{mass}} = 10^{-6}]$ respectively for each of the four objectives as were used in [98].

Termination was based on the maximum number of function evaluations that the MS and MP versions of the $\varepsilon$-NSGAII could use in a trial run. The termination criteria have been carefully designed based on previous studies [2, 97, 98] and problem difficulty while also considering limits on computing resources. In all applications, evaluations of speedup and performance metrics were based on 50 random trial runs for both the MS and MP versions of the $\varepsilon$-NSGAII. The performance of both parallelization strategies were analyzed for 1, 2, 4, 8, and 16 processor test cases to represent small-scale computing clusters and to maintain feasible queue times for the trial runs. The termination criteria are summarized in Table 4.1. For the DTLZ6 case study and the LTM case study, the $\varepsilon$ values of the $\varepsilon$-performance metric calculation were set to be equal to those of the algorithms $\varepsilon$-dominance values as described above. For the Leaf River test case, lower resolutions were used for the $\varepsilon$-performance metric calculation since many of the runs could not find reference solutions at $10^{-6}$ accuracy (see Table 4.1).

### 4.3 Results

Sections 4.3.1 through 4.3.3 present optimization results for each of the three case studies used to test the MS and MP versions of the $\varepsilon$-NSGAII. Each of the sections provides detailed tables of the average metric performance, run-time plots of search dynamics, and plots of success rates (defined in terms metric value goals). The
Table 4.1. Summary of parameter settings both MS and MP versions of the algorithm and metric calculations. NFE = Number of Function Evaluations; P = Number of Processors.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>DTLZ6</th>
<th>Leaf</th>
<th>LTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crossover Probability</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Mutation Probability</td>
<td>0.0833</td>
<td>0.077</td>
<td>0.04</td>
</tr>
<tr>
<td>Crossover Dist. Index</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>Mutation Dist. Index</td>
<td>20</td>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>$\varepsilon$ for Algorithm</td>
<td>$0.0045^* \times 10^{-6}$</td>
<td>$[1.0 \times 10^{-5} \ 10^{-2} \ 10^{-6}]$</td>
<td></td>
</tr>
<tr>
<td>Maximum NFE</td>
<td>$P \times 10^6$</td>
<td>$P \times 10^5$</td>
<td>$P \times 4 \times 10^5$</td>
</tr>
<tr>
<td>$\varepsilon$ for Metric Calc.</td>
<td>$0.0045^* \times 10^{-3}$</td>
<td>$[1.0 \times 10^{-5} \ 10^{-2} \ 10^{-6}]$</td>
<td></td>
</tr>
</tbody>
</table>

* the same $\varepsilon$ value is used for all the objectives

Mann-Whitney test [124] was used to validate if differences in the distributions of metric values attained at the same cutoff times for the MS and MP versions of the $\varepsilon$-NSGAII were statistically significant. The null hypothesis for the tests assumed that any two metric distributions were the same. In the results below, we report when there was at least a 95% confidence that the null hypothesis was rejected (i.e., the metric distributions are significantly different). The Mann-Whitney tests were used in two contexts: (1) inter-comparisons of the MS and MP versions of the $\varepsilon$-NSGAII at the same number of processors and (2) intra-comparisons that analyzed how increasing processor numbers improved the individual parallelization strategies relative to themselves.

4.3.1 Optimization results for case study 1: DTLZ6

Recall from Table 4.1 that the total number of function evaluations used to solve DTLZ6 was allowed to vary with increasing numbers of processors (i.e., NFE = $P \times 10^6$). The DTLZ6 problem is one of the most difficult test functions available in the EMO literature [2] and provides an excellent test of how problem difficulty impacts the efficiency and reliability of the MS and MP versions of the $\varepsilon$-NSGAII. Overall, Table (2) shows that the MP version of the $\varepsilon$-NSGAII had the best overall average metric values and that performance improved markedly with the increased search afforded by increased processors. All versions of the $\varepsilon$-NSGAII struggled to reliably solve DTZL6, particularly the MS strategy. Mann-Whitney inter-comparisons of
Table 4.2. DTLZ6 case study averages (AVG) and standard deviations (STD) in terms of the convergence (Conv.), diversity (Div.), $\varepsilon$-indicator (Eind.), and $\varepsilon$-performance (Eperf.) metrics. All metrics were computed using 50 random trials. MS and MP designate master-slave and multiple-population versions of the $\varepsilon$-NSGAII. The results are shown for 1, 2, 4, 8, and 16 processors. The best overall metrics are shown in underlined bold.

<table>
<thead>
<tr>
<th>#P</th>
<th>Strategy</th>
<th>Conv. ($\times 10^{-2}$)</th>
<th>Div. ($\times 10^{-2}$)</th>
<th>Eind. ($\times 10^{-2}$)</th>
<th>Eperf. ($\times 10^{-2}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Ideally = 0</td>
<td>Ideally = 1</td>
<td>Ideally = 0</td>
<td>Ideally = 1</td>
</tr>
<tr>
<td>1P</td>
<td>MS/MP</td>
<td>4.92</td>
<td>2.37</td>
<td>27.4</td>
<td>7.15</td>
</tr>
<tr>
<td></td>
<td>MS</td>
<td>4.92</td>
<td>2.37</td>
<td>27.6</td>
<td>7.01</td>
</tr>
<tr>
<td></td>
<td>MP</td>
<td>4.51</td>
<td>2.25</td>
<td>29.4</td>
<td>7.16</td>
</tr>
<tr>
<td>2P</td>
<td>MS</td>
<td>4.91</td>
<td>2.37</td>
<td>27.6</td>
<td>6.97</td>
</tr>
<tr>
<td></td>
<td>MP</td>
<td>3.29</td>
<td>2.27</td>
<td>34.1</td>
<td>8.71</td>
</tr>
<tr>
<td>4P</td>
<td>MS</td>
<td>4.91</td>
<td>2.37</td>
<td>27.5</td>
<td>6.98</td>
</tr>
<tr>
<td></td>
<td>MP</td>
<td>3.05</td>
<td>2.26</td>
<td>34.7</td>
<td>9.31</td>
</tr>
<tr>
<td>8P</td>
<td>MS</td>
<td>4.92</td>
<td>2.37</td>
<td>27.5</td>
<td>7.05</td>
</tr>
<tr>
<td></td>
<td>MP</td>
<td>2.21</td>
<td>1.54</td>
<td>36.2</td>
<td>8.46</td>
</tr>
<tr>
<td>16P</td>
<td></td>
<td></td>
<td></td>
<td>1.84</td>
<td></td>
</tr>
</tbody>
</table>

the MS and MP strategies showed that more than two processors are required for the methods to yield statistically meaningful performance differences for DTLZ6. In general, the MP strategy was superior to the MS approach for all performance metrics when four or more processors were used in search. The Mann-Whitney inter-comparison confirmed the MP strategy’s superiority at greater than a 97% confidence level.

The Mann-Whitney intra-comparisons indicated that an increase in the number of processors and the concomitant increase in NFE led to improved metrics’ distributions for either the MS or the MP versions of the $\varepsilon$-NSGAII. As indicated in Table 4.2 and confirmed with the Mann-Whitney intra-comparisons, increasing the number of processors used by the MS strategy did not yield statistically significant performance improvements. The implication of this result is that the serial version of the $\varepsilon$-NSGAII will not improve its ability to solve DTLZ6 with increases in search duration (i.e., using time continuation). This demonstrates that for extremely challenging problems where the serial version of the $\varepsilon$-NSGAII fails, the MS parallelization strategy will not improve search results.

Alternatively, the Mann-Whitney intra-comparisons for the MP version of the
Figure 4.6. Dynamic success rate plots for (a) the MS and (b) the MP configurations of the ε-NSGAII. Success rates are computed as the percentage of 50 trials that were able to attain an ε-performance value of 0.9. The success rates are shown as a function of computing clock time and processor count. Please note that 50 trial runs were used at each processor count.

ε-NSGAII showed statistically significant improvements in all metrics with increasing processor count. Figure (4.6) illustrates how processor count impacted the success rates for both the MS and MP versions of the ε-NSGAII. The ε-performance metric was used in Figure (4.6) because it best captured algorithmic reliability and showed similar dynamic trends compared to the other performance metrics. Figure (4.6) clearly shows that the MS parallelization strategy did not improve the ε-NSGAII’s success rate regardless of increasing processor count and search evaluations (e.g., 16 processors used 16 million function evaluations). Figure (4.6) shows that the MP parallelization strategy yielded more than a four-fold increase in success rate relative to the serial version of the ε-NSGAII. Mann-Whitney intra-comparisons for the MP strategy showed that increasing the processor count beyond eight did not yield statistically significant improvements in performance. Please note that speedup results were not presented for DTLZ6 since function evaluation time [i.e., $T_f$ in Eq. 4.2 and 4.3] was negligible compared to those used in the Leaf River and LTM case studies.
Table 4.3. Leaf River case study averages (AVG) and standard deviations (STD) in terms of the convergence (Conv.), diversity (Div.), ε-indicator (Eind.), and ε-performance (Eperf.) metrics. All metrics were computed using 50 random trials. MS and MP designate master-slave and multiple-population versions of the ε-NSGAII. The results are shown for 1, 2, 4, 8, and 16 processors. The best overall metrics are shown in underlined bold.

<table>
<thead>
<tr>
<th>#P</th>
<th>Strategy</th>
<th>Conv.($\times 10^{-2}$) AVG</th>
<th>Div.($\times 10^{-2}$) AVG</th>
<th>Eind.($\times 10^{-2}$) AVG</th>
<th>Eperf.($\times 10^{-2}$) AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Ideally = 0</td>
<td>Ideally = 1</td>
<td>Ideally = 0</td>
<td>Ideally = 1</td>
</tr>
<tr>
<td>1P</td>
<td>MS/MP</td>
<td>4.47</td>
<td>6.19</td>
<td>71.1</td>
<td>7.54</td>
</tr>
<tr>
<td></td>
<td>MS</td>
<td>3.55</td>
<td>5.84</td>
<td>77.5</td>
<td>8.74</td>
</tr>
<tr>
<td></td>
<td>MP</td>
<td>5.74</td>
<td>7.21</td>
<td>72.2</td>
<td>9.95</td>
</tr>
<tr>
<td>2P</td>
<td>MS</td>
<td>2.56</td>
<td>5.36</td>
<td>81.5</td>
<td>8.85</td>
</tr>
<tr>
<td></td>
<td>MP</td>
<td>2.04</td>
<td>3.64</td>
<td>80.1</td>
<td>5.52</td>
</tr>
<tr>
<td>4P</td>
<td>MS</td>
<td>1.88</td>
<td>4.81</td>
<td>86.1</td>
<td>8.96</td>
</tr>
<tr>
<td></td>
<td>MP</td>
<td>1.53</td>
<td>2.58</td>
<td>83.7</td>
<td>4.32</td>
</tr>
<tr>
<td>8P</td>
<td>MS</td>
<td>1.69</td>
<td>4.96</td>
<td>$\bf{91.3}$</td>
<td>9.08</td>
</tr>
<tr>
<td></td>
<td>MP</td>
<td>$\bf{0.96}$</td>
<td>0.17</td>
<td>86.9</td>
<td>4.40</td>
</tr>
</tbody>
</table>

### 4.3.2 Optimization results for case study 2: Leaf River calibration application

Tang et al. [97] used 100,000 model evaluations per ε-NSGAII trial run when calibrating the SAC-SMA hydrologic model of the Leaf River and showed that the algorithm had a relatively poor success rate. Recall that the Leaf River application is an unconstrained, continuous space problem with a potentially infinite solution space. Two key questions must be considered with the ε-NSGAII’s modest success rate: Is the Leaf River problem so difficult that the ε-NSGAII will always fail to reliably approximate the best known Pareto set, or, does the problem simply require much longer periods of search facilitated by the ε-NSGAII’s use of time-continuation? If true, the first question implies that the MP version of the ε-NSGAII should show superior performance as was observed for DTLZ6. Otherwise if the second question is true, than the MS version of the ε-NSGAII should be competitive if not superior to the MP version. Table [4.3] shows that in fact, the MS version of the ε-NSGAII was able to attain superior average scores for all of the performance metrics except convergence. When analyzing Table [4.3] recall that
Figure 4.7. Dynamic performance plots of ε-indicator for (a) the MS and (b) the MP versions of the ε-NSGAII. Each random trial is indicated with a solid line and the shaded regions show the ranges in performance. The ε-indicator values are shown as a function of computing clock time and processor count. Please note that 50 trial runs were used at each processor count.

the MS and MP versions of the ε-NSGAII used a total NFE equal to $P \times 10^5$ for the Leaf River case (where $P$ is processor count).

Mann-Whitney inter-comparisons between the MS and MP schemes for the Leaf River case study showed that performance differences were significant at greater than a 95% confidence level. Table 4.3 and the Mann-Whitney intra-comparisons show that increasing the processor count always improved the average performance metrics for the MS strategy. Distributional differences were validated at greater than a 98% confidence level for all of the performance metrics. The MP scheme required at least four processors to attain results that were statistically different from those attained with the serial version of the ε-NSGAII. Increasing the processor count beyond four processors, generally improved the MP scheme’s metric’s values at the 99% confidence level.

Figure (4.7) presents run-time ε-indicator dynamic results for the full distribution of random trials used at every processor count. The run-time dynamics provide a more detailed description of the MS and MP schemes’ dynamics and reliabilities. The results of all 50 random trial runs for each parallelization scheme
and processor count are shown in the figure. The $\varepsilon$-indicator metric plots are representative of the performance dynamics observed for all of the other performance metrics except convergence. Readers should exercise caution when interpreting or using the convergence metric because as its name implies scores only require a small number of points to be close to the reference set independent of how well they capture the full extent of its tradeoffs.

Generally both parallelization schemes have similar ranges in performance until greater than 8 processors were used. The individual trace lines for each trial run plotted in Figure (4.7) provide a more detailed understanding of why the MS strategy has better final average metrics. Although a small number of trials fail, the overall distribution of MS trials had dramatically better final metric values. Figure (4.7) highlights that the two parallelization strategies lead to very different search dynamics. Figures (4.7a) and (4.7b) are best interpreted by looking at the serial single processor search results and analyzing the influence of the increasing processor count. In particular, an increase from two to sixteen processors results in a clear shift in the search traces towards earlier time. These shifts towards earlier time represent the impacts of search speedup and indicate that improved $\varepsilon$-indicator values are being found earlier in time. Both Figures (4.7a) and (4.7b) show that perceived “failures” of the serial version of the $\varepsilon$-NSGAII are largely a function of time of search. As speedups increase, the percent of time dedicated to new search increases and the number of failures decreases.

Figures (4.8a) and (4.8b) clearly show that the MS scheme is able to attain high success rates more rapidly than the MP scheme with increasing processor counts. For example, for a processor count of 16 (i.e., NFE = $16 \times 10^5$) the MS strategy attains an 80 percent success rate in half of the time required by the MP scheme. In both figures (4.8a) and (4.8b) the general trend of increasing success rates in shorter time periods reflects the importance of speedup in allowing either parallelization scheme to actively search new regions of the Leaf River case study’s search space. Another key result shown in Figures (4.8a) and (4.8b) is that for water resources users constrained to using two-processor workstations, the MS strategy clearly outperforms MP search.

Judging speedup in parallel EMO applications is particularly challenging since solution quality must be considered explicitly. Solution quality judgments depend
Figure 4.8. Dynamic success rate plots for (a) the MS and (b) the MP configurations of the $\varepsilon$-NSGAII. Success rates are computed as the percentage of 50 trials that were able to attain an $\varepsilon$-indicator value of 0.1. The success rates are shown as a function of computing clock time and processor count. Please note that 50 trial runs were used at each processor count.

on the metrics being used. Figure (4.9) illustrates this issue for both parallelization strategies. In Figure (4.9), speedups were computed as the ratio of the average serial solution time $T_S$, versus the average parallel solution time $T_P$, required to attain each level of $\varepsilon$-indicator (see Eq. 4.1). When interpreting the results shown in Figure (4.9) it is important to remember what aspect of EMO performance is being captured by each of the performance metrics. The $\varepsilon$-indicator performance metric requires close proximity and a diverse distribution of solutions to attain near zero results. Figure (4.9a) indicates that the MS scheme’s final metric values were attained with linear or near linear speedups. In Figure (4.9b) the 8-processor and 16-processor speedups generally increase with more stringent metric values (i.e., as the metrics approach zero). Sub-linear speedups result from increased communication costs and the fact that the serial version of the $\varepsilon$-NSGAII quickly reduces the metric from 8 to 1 in the time allocated. Figure (4.9b) demonstrates that the MP version of the $\varepsilon$-NSGAII results in very different search dynamics and speedups. The most interesting result in the figure is that the speedup of the 16-processor case reaches 18 for $\varepsilon$-indicator metric. This result indicates that multiple populations increase search diversity and in some cases can in fact attain superlinear speedups. Note Figure (4.9b) also shows that the MP version of the $\varepsilon$-NSGAII exhibits poor speedups early and late in the runs. This result supports our observations in Figure (4.7), which show that reliably solving the Leaf
Figure 4.9. Joint plots of average speedup versus average solution quality for (a) the MS and (b) the MP configurations of the $\varepsilon$-NSGAII with respect to the $\varepsilon$-indicator metric. The averages are computed for 50 trial runs at each processor count. Speedups were computed as the ratio of the average serial time ($T_S$) versus the average parallel time ($T_P$) required to attain each level of $\varepsilon$-indicator.

River calibration problem requires sustained, high quality speedups to allow the $\varepsilon$-NSGAII to search new areas of the solution space. The poor speedups for the MP scheme can be attributed largely to the redundancy of search across processors and communication costs as has been observed in other studies [69].

4.3.3 Optimization results for case study 3: Long-term monitoring application

The LTM application represents an interesting contrast to the continuous, deceptive, and multimodal problem properties for the Leaf River calibration case study. The LTM application represents a discrete, non-deceptive constrained space that must be searched to identify a four-objective Pareto optimal set. Enumerative analysis has shown that nearly half of the decision space is infeasible and that the Pareto optimal set presents scaling challenges where objectives’ values range over several orders of magnitude [1, 98]. The average LTM solution evaluation time is approximately twice as long as solution evaluations for the Leaf River. Recall from Table 4.1 that both the MS and MP versions of the $\varepsilon$-NSGAII used a total NFE equal to $P \times 400,000$ (where $P$ is processor count). Table 4.4 provides the average
Table 4.4. LTM case study averages (AVG) and standard deviations (STD) in terms of the convergence (Conv.), diversity (Div.), $\varepsilon$-indicator (Eind.), and $\varepsilon$-performance (Eperf.) metrics. All metrics were computed using 50 random trials. MS and MP designate master-slave and multiple-population versions of the $\varepsilon$-NSGAIi. The results are shown for 1, 2, 4, 8, and 16 processors. The best overall metrics are shown in underlined bold.

<table>
<thead>
<tr>
<th>#P</th>
<th>Strategy</th>
<th>Conv.($\times 10^{-4}$)</th>
<th>Div.($\times 10^{-2}$)</th>
<th>Eind.($\times 10^{-2}$)</th>
<th>Eperf.($\times 10^{-2}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AVG</td>
<td>STD</td>
<td>AVG</td>
<td>STD</td>
</tr>
<tr>
<td>1P</td>
<td>MS/MP</td>
<td>6.74</td>
<td>1.27</td>
<td>79.8</td>
<td>1.87</td>
</tr>
<tr>
<td></td>
<td>MS</td>
<td>4.65</td>
<td>1.10</td>
<td>82.6</td>
<td>1.36</td>
</tr>
<tr>
<td></td>
<td>MP</td>
<td>4.42</td>
<td>0.90</td>
<td>83.3</td>
<td>1.66</td>
</tr>
<tr>
<td>2P</td>
<td>MS</td>
<td>2.98</td>
<td>0.89</td>
<td>85.1</td>
<td>1.02</td>
</tr>
<tr>
<td></td>
<td>MP</td>
<td>3.18</td>
<td>0.83</td>
<td>85.2</td>
<td>1.37</td>
</tr>
<tr>
<td>4P</td>
<td>MS</td>
<td>2.39</td>
<td>0.67</td>
<td>86.8</td>
<td>1.11</td>
</tr>
<tr>
<td></td>
<td>MP</td>
<td>2.72</td>
<td>0.78</td>
<td>86.6</td>
<td>0.92</td>
</tr>
<tr>
<td>8P</td>
<td>MS</td>
<td>2.16</td>
<td>0.60</td>
<td>88.4</td>
<td>1.07</td>
</tr>
<tr>
<td></td>
<td>MP</td>
<td>2.42</td>
<td>0.83</td>
<td>88.0</td>
<td>1.15</td>
</tr>
</tbody>
</table>

values of the performance metrics attained by both parallelization schemes for an increasing processor count. The table shows that the MS and MP schemes attained very similar average final metrics. Mann-Whitney inter-comparisons between the parallelization schemes showed that only the final $\varepsilon$-performance values’ statistical distributions were significantly different. The $\varepsilon$-performance metric is the most stringent of the metrics in terms of both convergence and diversity because the algorithms must find solutions that fall within very small $\varepsilon$-hyperboxes of the enumerated Pareto set for this test case. The Mann-Whitney intra-comparisons showed that increasing the processor count (and implicitly the NFE used) improved all metrics for both parallelization schemes with performance differences validated at greater than a 99% confidence level. The LTM results in Table 4.4 clearly show that the $\varepsilon$-NSGAIi’s performance is enhanced by the increased search afforded by parallelization, but they do not clearly differentiate the performances of the MS and MP strategies.

The run-time $\varepsilon$-performance plots presented in Figure 4.10 provide a more detailed description of the MS and MP strategies’ dynamics. Epsilon-performance was selected because this metric shows statistically significant performance differences between the MS and MP strategies. Although the final metrics for each of
the parallelization strategies are very similar, Figure (4.10) shows that they produced very different ranges of performance with increasing processor counts. The MS strategy clearly scales well with increasing numbers of processors as evidenced by its rapid and reliable run-time dynamics for the 8 and 16 processor cases. As was observed for the Leaf River case study, increased processor counts allow the MS version of the $\varepsilon$-NSGAII to better exploit time-continuation, resulting in the exploration of new regions of the decision space much earlier in the allocated 6000 seconds of run time.

Figure (4.11) further differentiates the MS and MP strategies’ performances using success rate plots. The success rate plots in Figure (4.11) show the cumulative
Figure 4.11. Dynamic success rates for (a) the MS and (b) the MP configurations of the $\varepsilon$-NSGAII. Success rates are computed as the percentage of 50 trials that were able to attain an $\varepsilon$-performance value of 0.8. The success rates are shown as a function of computing clock time and processor count.

distributions of the run times required to approximate 80-percent of the LTM case study’s Pareto optimal set. The steepness of the distributions gives a visual measure of the variance of run times (i.e., a perfectly vertical distribution would represent 50 trial runs with identical run times). Note that the serial results for the $\varepsilon$-NSGAII have been omitted from Figure (4.11) since none of the single processor trials satisfied our success criterion in 6000 seconds. Figures (4.10) and (4.11) show that two and four processor performance for the MS and MP strategies are very similar in terms of their reliability and speed. The largest performance differences between the MS and MP schemes resulted for the 8 and 16 processor trials. The MS strategy’s runtime distributions for processor counts of 8 and 16 are nearly vertical, which implies that their runtimes were nearly independent of random seed effects. Additionally, MS trials for 8 and 16 processors were completed two to three times faster than the MP trials for the same processor counts. The results shown in Figure (4.11) imply that the MS strategy is exploiting superior speedups to rapidly quantify the LTM case study’s Pareto surface.

Figure (4.12) confirms that the MS version of the $\varepsilon$-NSGAII was able to better sustain speedups over the full duration of runs. Figure (4.12a) shows that the MS scheme is able to attain nearly linear speedups for both performance metrics. The MS results exhibit the expected asymptotic limit on speedup (see Eqs. (4.1) and (4.2) with the increased communication costs associated with increasing processor
Figure 4.12. Joint plots of average speedup versus average solution quality for (a) the MS and (b) the MP configurations of the $\varepsilon$-NSGAII with respect to the $\varepsilon$-performance metric. The averages are computed for 50 trial runs at each processor count. Speedups were computed as the ratio of the average serial time ($T_S$) versus the average parallel time ($T_P$) required to attain each level of $\varepsilon$-performance.

Counts. Figures (4.12a) and (4.12b) show that the two and four processor speedups for both the MS and MP strategies are comparable as can also be seen in Figure (4.10). As shown in both Figures (4.12a) and (4.12b), although the MP version of the $\varepsilon$-NSGAII was able to enhance initial diversity which allowed it to identify feasible LTM solutions much more rapidly than the serial version of the $\varepsilon$-NSGAII, the MP strategy fails to sustain its search advantage when identifying solutions that are close in proximity and distributed uniformly over the LTM case study’s four-objective Pareto surface. Overall, the MS version of the $\varepsilon$-NSGAII exhibits superior performance on the LTM case study, especially when considering its simplicity and ease-of-implementation relative to the MP scheme.

4.4 Discussion

Generally, there are three primary factors that should be considered when evaluating parallelization strategies for multiobjective water resources applications: (1) problem difficulty, (2) the parallelization schemes’ ease-of-implementation, and (3) the parallel algorithm’s ease-of-use. The DTLZ6 test problem is potentially one of the hardest test functions available and has been shown to cause very high fail-
ure rates in all of the currently available benchmark EMO algorithms [2]. This is evident from our own analysis shown in Figure 4.4. The DTLZ6 problem serves two purposes in this study: (1) it provides a baseline upper bound for analyzing how problem difficulty impacts the relative performances of the MS and MP parallelization schemes and (2) it validates that the MP version of the \(\varepsilon\)-NSGAII is more effective than the MS scheme when solving extremely difficult problems. The success rates shown in Figure 4.6 show that the MP parallelization scheme is far superior when solving DTLZ6. As has been highlighted in prior studies [35, 72], the MP scheme significantly changes solution diversity and search dynamics, whereas the MS parallelization strategy only changes the duration of search. Although our dynamic deme-sizing and migration strategies simplify the parameterization requirements of the MP version of the \(\varepsilon\)-NSGAII, the parallelization scheme still requires a much higher degree of user sophistication than does the MS strategy. Multiple-population EMO algorithms take considerably more effort to implement and yield dramatically more complex speedup dynamics relative to MS schemes. Although the MP version of the \(\varepsilon\)-NSGAII was required to solve the DTLZ6 problem, the interesting issue that water resources scientists and engineers need to consider is: How many water resources applications are as difficult as DTLZ6?

This question is particularly relevant considering that the MP version of the \(\varepsilon\)-NSGAII was inferior to the MS scheme for the Leaf River and LTM applications. The Leaf River and the LTM test cases encompass two very different problem types from the water resources literature that have been shown to be challenging for modern EMO algorithms (e.g., see [53, 97, 98]). Figures 4.8 and 4.11 show that the serial version of the \(\varepsilon\)-NSGAII had very low success rates when solving the Leaf River and the LTM test cases (defined in terms of attaining a goal performance metric value within a user defined runtime). In water resources applications where high EMO algorithm failure rates are observed, two key questions arise: Are these problems so difficult that the EMO algorithm will always fail to reliably approximate the best known Pareto set, or, do the problems simply require much longer periods of search facilitated by parallelization and time-continuation?

Sections 4.3.2 and 4.3.3 show that enhanced search durations and time-continuation allowed the MS version of the \(\varepsilon\)-NSGAII to rapidly and reliably improve search success rates. In the serial and both of the parallel versions of
the \(\varepsilon\)-NSGAI\(I\)I, time continuation results from combining random solutions with \(\varepsilon\)-dominance archive members each time the population size is adapted. Each new population is equal to four times the current \(\varepsilon\)-dominance archive’s size so that \(\varepsilon\)-nondominated archive members compose 25-percent of the new population and the remaining 75-percent is composed of randomly generated solutions. Time continuation allows the algorithm to maintain effective search for as long as is necessary or is computationally tractable. The \(\varepsilon\)-NSGAI\(I\)I’s failure rates for the Leaf River and LTM test cases shown in Figures 4.8 and 4.11 result from time constraints and not from algorithmic limitations, which is often the case for water resources applications (e.g., see [158, 159]).

In both the Leaf River and LTM test cases, linear to near linear speedups allowed the MS version of the \(\varepsilon\)-NSGAI\(I\)I to better exploit time-continuation, resulting in the exploration of new regions of their decision spaces much earlier in the allocated run times. The MP version of the \(\varepsilon\)-NSGAI\(I\)I has significantly more complex speedup dynamics relative to the MS version. For extremely difficult problems, the MP scheme’s redundancy in search and its enhanced diversity from multiple independently searching populations dramatically improves its search capabilities relative to the single-population versions of the \(\varepsilon\)-NSGAI\(I\)I. Ironically, the strengths of the MP scheme that emerge for very difficult problems cause it to be inferior to the single population MS version of the \(\varepsilon\)-NSGAI\(I\)I for less difficult problems where extended search times will reliably yield high quality results. In other words, it takes much longer to converge multiple populations to Pareto optimal fronts and the MP scheme’s complex speedup dynamics do not guarantee extended search periods.

A contribution of this research is to demonstrate that the \(\varepsilon\)-NSGAI\(I\)I’s auto-adaptive population sizing, \(\varepsilon\)-dominance archiving, and time continuation when combined with a simple MS strategy can yield superior search relative to MP strategies. Additionally, as the solution evaluation times for water resources applications increases, the scalability of the MS version of the \(\varepsilon\)-NSGAI\(I\)I will improve (i.e., the asymptotic limits on MS speedups that result from communication costs decrease). Readers should also note the importance of monitoring solution quality using multiple EMO performance metrics, especially when performing speedup calculations. Overall, the MS version of the \(\varepsilon\)-NSGAI\(I\)I exhibits superior perfor-
mance on both of the water resources applications, especially when considering its simplicity and ease-of-implementation relative to the MP scheme.

4.5 Conclusions

This chapter uses a formal metrics-based framework to demonstrate the use of the MS and MP parallelization schemes with the $\varepsilon$-NSGAII. The MS and MP versions of the $\varepsilon$-NSGAII generalize the algorithm’s auto-adaptive population sizing, $\varepsilon$-dominance archiving, and time continuation to a distributed processor environment. A key finding of this work is that time-continuation and parallel speedups can dramatically improve the efficiency and reliability of EMO algorithms in water resources applications. Time continuation is an evolutionary algorithm search enhancement that promotes solution diversity and allows the $\varepsilon$-NSGAII to maintain effective search for as long as is necessary or is computationally tractable. This chapter uses three test cases to compare the MS and MP versions of the $\varepsilon$-NSGAII: (1) an extremely difficult benchmark test function termed DTLZ6 drawn from the computer science literature, (2) an unconstrained continuous hydrologic model calibration test case for the Leaf River near Collins, Mississippi, and (3) a discrete, constrained, four-objective long-term groundwater monitoring (LTM) application. These test cases were carefully selected to encompass a broad range of problem properties and clearly demonstrate the importance of problem difficulty when selecting parallelization strategies for multiobjective applications.

The MP version of the $\varepsilon$-NSGAII is more effective than the MS scheme when solving DTLZ6. This chapter highlights that artificially constructed and extremely difficult test problems such as DTLZ6 may wrongly bias water resources applications towards using more complicated algorithms (i.e., the MP parallelization scheme), when a simpler MS strategy may work as well or better. Overall, the MS version of the $\varepsilon$-NSGAII exhibits superior performance on both of the water resources applications, especially when considering its simplicity and ease-of-implementation relative to the MP scheme. This chapter also clearly demonstrates the importance of monitoring solution quality using multiple EMO performance metrics, especially when performing speedup calculations. Overall, the three test cases used in this chapter demonstrate that the MS and MP versions of the $\varepsilon$-
NSGAII can broaden the size and difficulty of multiobjective water resources applications that can be solved efficiently and reliably. Future extensions of this work should explore how heterogeneous computer networks impact the relative benefits of the MS and MP parallelization strategies.
Study 3: Assessing sensitivity analysis methods to advance lumped watershed model identification and evaluation

This chapter is drawn from Y. Tang et al.’s paper that is in review in the journal of Hydrology and Earth System Science [160]. Chapter 3 and 4 show that hydrologic model calibration problems pose significant challenges in terms of their computational demands and their complexity. Chapter 4 addresses the computational demands posed by multiobjective calibration through novel parallelization schemes for the evolutionary multiobjective optimization (EMO) algorithms. This chapter and Chapter 6 focus on sensitivity analysis (SA) as a means to improve our understanding of model behavior and simplify problem formulations when calibrating complex models. This chapter tested four sensitivity analysis methods: (1) local analysis using parameter estimation software (PEST), (2) regional sensitivity analysis (RSA), (3) analysis of variance (ANOVA), and (4) Sobols method. These four methods were tested to identify sensitivity tools that will advance our understanding of lumped hydrologic models for the purposes of model improvement, calibration efficiency and improved measurement schemes. The methods’ relative efficiencies and effectiveness have been analyzed and compared. These four sen-
sitivity methods are applied to the lumped Sacramento soil moisture accounting model (SAC-SMA) coupled with SNOW-17.

5.1 Introduction

This chapter evaluates the differences between four popular sensitivity analysis methods, selected to represent the variety of methods currently used. The four sensitivity analysis methods include: (1) PEST, (2) RSA, (3) ANOVA, and (4) Sobols method. The methods are applied to the Sacramento soil moisture accounting model, a medium complexity spatially lumped rainfall-runoff model used for river forecasting throughout the USA (the same lumped model used in Chapters 3 and 4). The model is implemented in two watersheds in the Susquehanna River Basin in Pennsylvania and run at hourly, six hourly, and daily time steps.

Broadly, models of watershed hydrology are irreplaceable components of water management studies including flood and drought prediction, water resource assessment, climate and land use change impacts, or non-point source pollution analysis [e.g., 161]. Hydrologic models are evolving from single purpose tools to complex decision support systems that can perform all (or at least many) of the tasks mentioned above in a single software package. Hydrological models vary in complexity from lumped conceptual models to distributed models that include close coupling of surface and groundwater flow processes, feedbacks with the atmosphere, transport of water and solutes, and spatially explicit representations of system characteristics and states [e.g., 6, 115, 162, 163]. In integrated assessment applications models may even include socioeconomic components to integrate human behavior [164]. In general, hydrologic models are highly non-linear, contain thresholds, and often have significant parameter interactions. These properties make it difficult to evaluate how models of hydrologic systems behave and which parameters control this behavior during different response modes [e.g., 165]. The increasing trend towards more complex models and its potential consequences in terms of computational constraints and obfuscating model impacts on decision making motivates the need for enhanced model identification and evaluation tools [12, 53, 75, 80].

Hydrologic models play an important role in elucidating the dominant controls
on watershed behavior and in this context it is important for hydrologists to identify the dominant parameters controlling model behavior. One approach to gain this understanding is through the use of sensitivity analysis, which evaluates the parameter’s impacts on the model response [24–30]. Sensitivity analysis results can be used to decide which parameters should be the focus of model calibration efforts, or even as an analysis tool to test if the model behaves according to underlying assumptions [e.g., [31]]. Ultimately, sensitivity methods should serve as diagnostic tools that help to improve mathematical models and potentially help us to identify where gaps in our knowledge are most severe and are most strongly affecting prediction uncertainty. Data gaps are particularly important in the context of guiding field measurement campaigns [80, 166–168]. Section 5.2 provides a more detailed review of existing sensitivity analysis methods and a detailed discussion of the four methods compared in this chapter.

5.2 Sensitivity analysis tools and sampling schemes

5.2.1 Overview

Model sensitivity analysis characterizes the impact that changes in model inputs have on the model outputs in a strict sense. Model inputs include model parameters, forcing, initial conditions, boundary conditions, etc. In this study, we focus on analyzing the sensitivities of model parameters. Sensitivity measures are determined mathematically, statistically, or even graphically. There are several prior studies that have broadly reviewed and classified the sensitivity analysis methods that exist [74, 79]. Any sensitivity analysis approach can be broken up into two components [80]: (1) a strategy for sampling the model parameter space (and/or state variable space), and (2) a numerical or visual measure to quantify the impacts of sampled parameters on the model output of interest. The implementation of these two components varies immensely [e.g., [25, 29, 78, 81, 83]], and guidance is currently lacking to help modelers decide which approach is best suited to the needs of a particular study. Generally, the approaches can be categorized into two main groups–local methods and global methods [22, 84].
The nominal range and differential analysis methods are two well known local parameter sensitivity analysis methods \cite{76, 78}. Nominal range sensitivity analysis calculates the percentage change of outputs due to the change of model inputs relative to their baseline (nominal) values. The percentage change is seen as the sensitivity of the corresponding input. Differential analysis utilizes partial derivatives of the model outputs with respect to the perturbations of the model input. The derivative values are themselves the metrics of sensitivity. Further analysis can be conducted by approximating the simulation model using Taylor’s series \cite{76}.

The nominal range and differential analysis methods have the advantages of being straightforward to implement while maintaining modest computational demands. The major drawback of these methods is their inability to account for parameter interactions, making them prone to underestimating true model sensitivities. Alternatively, global parameter sensitivity analysis methods vary all of a model’s parameters in predefined regions to quantify their importance and potentially the importance of parameter interactions.

There are a variety of global sensitivity analysis methods such as regional sensitivity analysis (RSA) \cite{24, 87}, variance based methods \cite{74}, regression based approaches \cite{88, 89}, and Bayesian sensitivity analysis \cite{77}. Global methods attempt to explore the full parameter space within pre-defined feasible parameter ranges. In this chapter, our goal is to test a suite of sensitivity methods and discuss their relative benefits and limitations for advancing lumped watershed model identification and evaluation.

The four sensitivity analysis approaches include one local method termed PEST and three global methods consisting of RSA, analysis of variance (ANOVA), and Sobol’s method. These sensitivity analysis methods were selected for comparison due to their popularity and their common application in a variety of scientific domains \cite{25, 28, 29, 31, 32, 90, 96}. The sensitivity analysis methods tested in this chapter range from local to global and capture a broad range of analysis methodologies (differential analysis, RSA, and variance-based analysis). The main characteristics of these four methods are summarized in Table 5.1 \cite{5.1}. In Section 5.2.2 each of these approaches and the associated statistical sampling schemes used in this chapter are discussed in more detail. In the context of this chapter we assume that the selection of an appropriate numerical measure, is satisfied through two
chosen objective functions based on the root mean square error (RMSE) (see Section 5.5.2). Readers interested in how parameter sensitivity changes with different objective functions can reference the following studies [26, 165].

### 5.2.2 Sensitivity analysis tools

#### 5.2.2.1 PEST

PEST, which stands for parameter estimation, is a model independent nonlinear parameter estimation tool [32, 86, 90, 91]. PEST was developed to facilitate data interpretation, model calibration and predictive analysis. Like many other parameter estimation or model calibration tools, PEST aims to match the model simulation with an observed set of data by minimizing the weighted sum of squared differences between the two. The optimization problem is iteratively solved by linearizing the relationship between a model’s output and its parameters. The linearization is conducted using a Taylor series expansion where the partial derivatives of each model output with respect to every parameter must be calculated at every iteration. For each iteration, the solution of the linearized problem is the current optimal set of parameters. The current optimal set is then compared to that of the previous time step to determine when to terminate the optimization process. During the linearization step, the forward difference or central difference operators can be used for calculating the derivatives. Parameter ranges, initial parameter values, and parameter increments must be provided by the user. The parameter vector is updated at each step using the Gauss-Marquardt-Levenberg algorithm [169, 170]. The derivatives of the model outputs with respect to its parameters are calculated and provide a measure of the parameter sensitivities at each iteration. The “composite sensitivity” is provided by PEST as a byproduct of the parameter estimation results. The composite sensitivity of parameter $i$ is...
defined as:

\[ s_i = (J^tQJ)_{ii}^{1/2} / m \]  

(5.1)

where \( J \) is the Jacobean matrix and \( Q \) is the cofactor matrix which in most cases is a diagonal matrix whose elements are composed of squared weights for model outputs. If the model outputs are equally weighted, \( Q \) is equal to the identity matrix. The number of outputs, \( m \), is the number of data records in the time series in this study. Thus \( s_i \) is the normalized magnitude of the Jacobean matrix column with respect to parameter \( i \). As expected for a local sensitivity analysis method, Eq. 5.1 is a univariate analysis of parameter impacts on model outputs (i.e., no parameter interactions are considered).

### 5.2.2.2 Regional sensitivity analysis using Latin hypercube sampling

RSA \([24, 87]\) is also called generalized sensitivity analysis (GSA) \([25]\) and has been widely used in hydrology [e.g. \([25, 29, 30, 88, 92, 171]\)]. Monte Carlo sampling and “behavioral/nonbehavioral” partitioning are the two major components of this method. Monte Carlo sampling is used to generate \( n \) parameter sets in the feasible parameter space defined using a multi-variate uniform distribution. After model evaluations using these parameters, the sets of parameters are decomposed into two separate groups (behavioral/good and nonbehavioral/bad) according to the model’s performance or behavior. RSA identifies the difference between the underlying distributions of the behavioral and nonbehavioral groups. Either graphical methods (e.g., marginal cumulative distribution function plots) or statistical methods such as Kolmogorov-Smirnov (KS) testing \([172]\) are then used to characterize if a parameter significantly impacts behavioral results.

Freer et al. \([25]\) extended the original RSA by breaking the behavioral parameter sets into ten equally sized groups. Wagener et al. \([26]\) modified this approach further by including all parameter sets and avoiding the need to specify behavioral and non-behavioral sets. Instead, the population is divided into ten bins of equal size based on a sorted model performance measure \([80]\). Conclusions about parameter sensitivities are made qualitatively by examining differences in the marginal cumulative distributions of a parameter within each of the ten groups. The ten lines in the RSA plot represent the cumulative distributions of a parameter with
respect to ten sampled sub-ranges. If the lines are clustered, the parameter is not sensitive to a specific model performance measure. Conversely, the degree of dispersion of the lines is a visual measure of a model’s sensitivity to an input parameter. Wagener and Kollat [80] implemented the original idea of Freer et al. [25] visually using the Monte Carlo analysis toolbox (MCAT) [26, 31, 173] where the marginal cumulative distributions of the ten groups are plotted as the likelihood value versus the parameter values.

In this study, Latin hypercube sampling (LHS) was used to sample the feasible parameter space for testing RSA based on the recommendations and findings of prior studies [e.g., 30, 174]. LHS integrates random sampling and stratified sampling [76, 175] to make sure that all portions of the parameter space are considered. The method divides the parameters’ ranges into \( n \) disjoint intervals with equal probability \( 1/n \) from which one value is sampled randomly in each interval. LHS is generally recommended for sparse sampling of the parameter space and the parameter interactions are neglected as noted by [176]. More details about LHS are available in the following papers [76, 175, 176].

5.2.2.3 Analysis of variance using iterated fractional factorial design sampling

Assuming model response (e.g., RMSE of streamflow in this study) is normally distributed, the role of ANOVA is to quantify the differences of the mean model responses that result from samples of each parameter. In ANOVA, parameters are “grouped” into particular ranges of parameter values representing intervals with equal parameter value width, contrasting to RSA in which parameter sets are “grouped” based on model response measures such as the RMSE of streamflow predictions used in this study. According to ANOVA terminology, a parameter is called a “factor” and a parameter group is termed a “level” of the factor. ANOVA essentially partitions the model output or response into the overall mean, main factor effects, factor interactions, and an error term [93, 177]. Theoretically, ANOVA can capture a range from the first order (main effects from single parameters) to the total order of effects (i.e., all parameter impacts including all interactions). However, it is not feasible to calculate all of the effects for a complex model in practice due to computational limitations. Fortunately, prior studies have shown
that second order interactions are usually sufficient for capturing a model’s output variance \cite{27, 178, 179}. Therefore, our analysis focuses on first order and second order effects within the ANOVA model. The model response variable $Y$ is decomposed into main and second order effects of two factors according to 2-way ANOVA model:

$$Y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha \times \beta)_{ij} + \varepsilon_{ijk}$$  \hspace{1cm} (5.2)

where $i$ and $j$ indicate the levels of factors $A$ and $B$ respectively, $\alpha_i$ is the main effect of $i^{th}$ level of $A$, $\beta_j$ is the main effect of $j^{th}$ level of $B$, $(\alpha \times \beta)_{ij}$ represents the interaction of $A$ and $B$. The error term, $\varepsilon_{ijk}$, reflects the effects that are not explained by the main effects and interactions of the two factors. Variable $k$ represents the $k^{th}$ value of $Y$.

The $F$-test is used to evaluate the statistical significance of differences in the mean responses among the levels of each parameter or parameter interaction. The $F$ values are calculated for all parameters and parameter interactions. The higher the $F$ values are, the more significant the differences are and therefore the more sensitive the parameter or parameter interaction is. Detailed presentation of the ANOVA calculation table for main effects and second order effects can be found in other studies \cite{93, 177}. In addition to the $F$-test, the coefficient of determination ($r^2$) quantifies how the ANOVA model shown in Eq. (5.2) captures the total variation of model responses with the inclusion of the second order parameter interactions. In cases where parameter interactions are important the coefficient of determination should improve (or increase) with the addition of the interaction term $(\alpha \times \beta)_{ij}$ from Eq. (5.2).

When applying the ANOVA method the statistical sampling scheme used to quantify the model response is a key determinant of the method’s computational feasibility and accuracy. If one parameter or parameter interaction is analyzed at a time in succession, the total number of model runs will be excessively large and most hydrologic applications would be computationally intractable. In this study, the iterated fractional factorial design (IFFD) sampling scheme \cite{180, 182} was used to limit the computational burden posed by ANOVA while seeking high quality results.

IFFD works well when first and second order parameter effects dominate \cite{181}. Using IFFD in ANOVA allows users to neglect higher order interactions not in-
cluded in the model \cite{27, 181} while generating highly repeatable results \cite{182}. Consequently, the number of model runs required can be reduced substantially. IFFD as implemented in this study samples the parameters at three different levels: low, middle, and high. The parameter levels are defined as equally spaced intervals within the predefined parameter ranges. Using a small number of factor levels enables the sampling scheme to attain statistically significant results efficiently and accurately \cite{93, 181}. IFFD extends the basic orthogonal fractional factorial design (FFD) by conducting multiple iterations. The basic operations in IFFD include orthogonalization, folding, replication and random sampling \cite{180–182}. The orthogonalized design guarantees equal frequency for two parameter combinations but also differentiates the main effects from two-way interactions. A detailed presentation of IFFD is beyond the scope of this chapter. Readers interested in detailed descriptions of IFFD are referred to the following papers \cite{180–182}.

5.2.2.4 Sobol’s method using quasi-random sequence sampling

In Sobol’s method \cite{94}, the variance of the model output is decomposed into components that result from individual parameters as well as parameter interactions. Conventionally, the direct model output is replaced by a model performance measure such as RMSE as used in this study. The sensitivity of each parameter or parameter interaction is then assessed based on its contribution (measured as a percentage) to the total variance computed using a distribution of model responses. Assuming the parameters are independent, the Sobol’s variance decomposition is:

\[
D(y) = \sum_i D_i + \sum_{i<j} D_{ij} + \sum_{i<j<k} D_{ijk} + D_{12...m}
\]  \hspace{1cm} (5.3)

where \( D_i \) is the measure of the sensitivity to model output \( y \) due to the \( i^{th} \) component of the input parameter vector denoted as \( \Theta_i \), \( D_{ij} \) is the portion of output variance that results due to the interaction of parameter \( \theta_i \) and \( \theta_j \). The variable \( m \) defines the total number of parameters. The variance decomposition shown in Eq. \( 5.3 \) can be used to define the sensitivity indices of different orders as:

\[
\text{first order } S_i = \frac{D_i}{D} \hspace{1cm} (5.4)
\]
second order \( S_{ij} = \frac{D_{ij}}{D} \)  \hspace{1cm} (5.5)

total \( S_{T_i} = 1 - \frac{D_{\sim i}}{D} \)  \hspace{1cm} (5.6)

where \( S_i \) denotes the sensitivity that results from the main effect of parameter \( \theta_i \). The second order sensitivity index, \( S_{ij} \), defines the sensitivity that results from the interaction of parameters \( \theta_i \) and \( \theta_j \). The average variance, \( D_{\sim i} \), results from all of the parameters except for \( \theta_i \). The total order sensitivity, \( S_{T_i} \), represents the main effect of \( \theta_i \) as well as its interactions up to \( m^{th} \) order of analysis. A parameter which has a small first order index but large total sensitivity index primarily impacts the model output through parameter interactions.

The variances in Eq. 5.3 can be evaluated using approximate Monte Carlo numerical integrations. The Monte Carlo approximations for \( D, D_i, D_{ij} \), and \( D_{\sim i} \) are defined as presented in the following prior studies [28, 94, 95]:

\[
\hat{f}_0 = \frac{1}{n} \sum_{s=1}^{n} f(\Theta_s) \quad (5.7)
\]

\[
\hat{D} = \frac{1}{n} \sum_{s=1}^{n} f^2(\Theta_s) - \hat{f}_0^2 \quad (5.8)
\]

\[
\hat{D}_i = \frac{1}{n} \sum_{s=1}^{n} f(\Theta_s^{(a)}) f(\Theta_s^{(b)}_{(\sim i) s}, \Theta_s^{(a)}) - \hat{f}_0^2 \quad (5.9)
\]

\[
\hat{D}_{ij}^c = \frac{1}{n} \sum_{s=1}^{n} f(\Theta_s^{(a)}) f(\Theta_s^{(b)}_{(\sim i, \sim j) s}, \Theta_s^{(a)}_{(i,j) s}) - \hat{f}_0^2 \quad (5.10)
\]

\[
\hat{D}_{ij} = \hat{D}_{ij}^c - \hat{D}_i - \hat{D}_j \quad (5.11)
\]

\[
\hat{D}_{\sim i} = \frac{1}{n} \sum_{s=1}^{n} f(\Theta_s^{(a)}) f(\Theta_s^{(a)}_{(\sim i) s}, \Theta_s^{(b)}) - \hat{f}_0^2 \quad (5.12)
\]

where \( n \) is the sample size, \( \Theta_s \) denotes the sampled individual in the scaled unit hypercube, and \( (a) \) and \( (b) \) are two different samples. All of the parameters take their values from sample \( (a) \) are represented by \( \Theta_s^{(a)} \). The variables \( \Theta_s^{(a)} \) and \( \Theta_s^{(b)} \) denote that parameter \( \theta_i \) uses the sampled values in sample \( (a) \) and \( (b) \), respectively. The symbols \( \Theta_s^{(b)}_{(\sim i) s} \) and \( \Theta_s^{(b)}_{(\sim i) s} \) represent cases when all of the parameters except for \( \theta_i \) use the sampled values in sample \( (a) \) and \( (b) \), respectively. The symbol \( \Theta_s^{(a)}_{(i,j) s} \) represents parameters \( \theta_i \) and \( \theta_j \) with sampled values in sample \( (a) \). Finally,
\( \Theta^{(a)}_{(\sim i, \sim j)} \) represents the case when all of the parameters except for \( \theta_i \) and \( \theta_j \) utilize sampled values from sample \((b)\).

The original Sobol’s method required \( n \times (2m + 1) \) model runs to calculate all the first order and the total order sensitivity indices. An enhancement of the method made by Saltelli [183] provides the first, second and total order sensitivity indices using \( n \times (2m + 2) \) model runs. In this study, we implemented this modified version of Sobol’s methodology to compute the first order, second order and total order indices.

The convergence of the Monte Carlo integrations used in Sobol’s method is heavily affected by the sampling scheme selected. The error term in the Monte Carlo integration decreases as a function of \( 1/\sqrt{n} \) given uniform, random samples at \( n \) points in the \( m \)-dimensional space. However, in this study we elected to use Sobol’s quasi-random sequence [184-185] to increase the convergence rate to nearly \( 1/n \). The quasi-random sequence samples points more uniformly along the Cartesian grids than uncorrelated random sampling. Details about Sobol’s quasi-random sequence can be found in the following studies [176, 184-186].

### 5.3 Overview of the lumped hydrologic models

The SNOW-17 [187] and the Sacramento soil moisture accounting (SAC-SMA) models [113] are popular and the United States National Weather Service (US NWS) uses them for river forecasting [163, 188-190]. In this chapter, lumped versions of these models have been coupled where SAC-SMA uses SNOW-17’s outputs as forcing. Sections [5.3.1] and [5.3.2] provide brief overviews of both models.

#### 5.3.1 SNOW-17

SNOW-17 [187] is a conceptual model that simulates the energy balance of a snowpack using a temperature index method. Air temperature and precipitation are the model inputs. The states and processes include snow melt, snow cover accumulation, surface energy exchange during non-melt periods, snow cover heat storage, areal extent of snow cover, retention and transmission of liquid water, and heat exchange at the snow-soil interface. Snow melt, snow cover accumulation, and
Figure 5.1. Major SNOW-17 processes and their corresponding parameters. MBASE-Base temperature for snowmelt computations during nonrain periods (degc). NMF-Maximum negative melt factor (mme/degc/6hr). TIPM-Antecedent temperature index parameter. PLWHC-Percent (decimal) liquid-water holding capacity. DAYGM-Constant amount of melt which occurs at the snow-soil interface whenever snow is present (mm). The full description of other parameter names can be found in section 5.3.1 and in Table 5.2. Shaded boxes represent the states or processes.

Areal extent are the three most influential components in the model.

Snow melt is calculated separately for rain-on-snow periods and non-rain periods. The snow melt during rain-on-snow periods is computed based on energy and mass balance equations with average wind function (UADJ) as the only parameter. In contrast, snow melt during non-rain periods is calculated empirically. The maximum melt factor (MFMAX) and the minimum melt factor (MFMIN) control this calculation. When calculating the accumulation of snow cover, the form of precipitation is simply determined by a threshold temperature (PXTEMP). The snowfall correction factor (SCF) adjusts gage precipitation estimates for biases during snowfall. To determine the areal extent of snow cover, a pre-defined depletion curve relates the areal extent to areal water equivalent based on the historical maximum water equivalent and the water equivalent above which 100% of snow cover exists. Process calculations are described in more detail in Anderson [187]. The main processes and corresponding twelve model parameters in SNOW-17 are shown in Fig. 5.1. Based on the prior work of Anderson [191], we have focused our sensitivity analysis on five of SNOW-17’s parameters (excluding the areal depletion curve). These five parameters and their allowable ranges [191] are summarized in Table 5.2.
### Table 5.2. Summary of SNOW-17 and SAC-SMA parameters

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameters</th>
<th>Unit</th>
<th>Description</th>
<th>Allowable Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNOW-17</td>
<td>SCF</td>
<td>gage catch deficiency adjustment factor</td>
<td></td>
<td>1.0-1.3</td>
</tr>
<tr>
<td></td>
<td>MFMAX</td>
<td>mm/°C/6h</td>
<td>Maximum melt factor during non-rain periods</td>
<td>0.1-1.2</td>
</tr>
<tr>
<td></td>
<td>MFMIN</td>
<td>mm/°C/6h</td>
<td>Minimum melt factor during non-rain periods</td>
<td>0.1-0.6</td>
</tr>
<tr>
<td></td>
<td>UADJ</td>
<td>mm/mb/6h</td>
<td>Average wind function during rain-on-snow periods</td>
<td>0.02-0.2</td>
</tr>
<tr>
<td></td>
<td>SI</td>
<td>mm</td>
<td>Mean water-equivalent above which 100% cover exists</td>
<td>10-120</td>
</tr>
<tr>
<td></td>
<td>UZTWM</td>
<td>mm</td>
<td>Upper zone tension water maximum storage</td>
<td>1.0-150.0</td>
</tr>
<tr>
<td></td>
<td>UZKW</td>
<td>mm</td>
<td>Upper zone free water maximum storage</td>
<td>1.0-150.0</td>
</tr>
<tr>
<td></td>
<td>UZK</td>
<td>day⁻¹</td>
<td>Upper zone free water lateral depletion rate</td>
<td>0.1-0.5</td>
</tr>
<tr>
<td></td>
<td>PCTIM</td>
<td></td>
<td>Impervious fraction of the watershed area</td>
<td>0.0-0.1</td>
</tr>
<tr>
<td></td>
<td>ADIMP</td>
<td></td>
<td>Additional imperious area</td>
<td>0.0-0.4</td>
</tr>
<tr>
<td></td>
<td>ZPERC</td>
<td></td>
<td>Maximum percolation rate</td>
<td>1.0-250.0</td>
</tr>
<tr>
<td>SAC-SMA</td>
<td>REXP</td>
<td></td>
<td>Exponent of the percolation equation</td>
<td>0.0-5.0</td>
</tr>
<tr>
<td></td>
<td>LZTWM</td>
<td>mm</td>
<td>Lower zone tension water maximum storage</td>
<td>1.0-500.0</td>
</tr>
<tr>
<td></td>
<td>LZFWM</td>
<td>mm</td>
<td>Lower zone free water supplemental maximum storage</td>
<td>1.0-1000.0</td>
</tr>
<tr>
<td></td>
<td>LZFPM</td>
<td>mm</td>
<td>Lower zone free water primary maximum storage</td>
<td>1.0-1000.0</td>
</tr>
<tr>
<td></td>
<td>LZSK</td>
<td>day⁻¹</td>
<td>Lower zone supplemental free water depletion rate</td>
<td>0.01-0.25</td>
</tr>
<tr>
<td></td>
<td>LZPK</td>
<td>day⁻¹</td>
<td>Lower zone primary free water depletion rate</td>
<td>0.0001-0.025</td>
</tr>
<tr>
<td></td>
<td>PFREE</td>
<td></td>
<td>Fraction of water percolating from upper zone directly to lower zone free water storage</td>
<td>0.0-0.6</td>
</tr>
</tbody>
</table>

#### 5.3.2 Sacramento soil moisture accounting model

The SAC-SMA model [113] is a sixteen parameter lumped conceptual watershed model used for operational river forecasting by the US NWS. It represents the soil column by an upper and lower zone of multiple storages. The upper zone is divided into free water and tension water storages. The tension water spills into the free water storage only when the tension water storage (UZTWM) is filled. The free water in the upper zone can then move laterally as interflow or move vertically down to the lower zone as percolation. Capacities of the two storages are model parameters (UZFWM and UZTWM), while the volume of water in each at any time step are model states. Similar to the upper zone, the lower zone also has tension water and free water storages. The free water in the lower zone is further partitioned into two types: primary and supplemental free water storages, both of which can contribute to base-flow but drain independently at different speeds following Darcy’s law. The maximum storages for these different types of lower zone water are the lower zone maximum tension water (LZTWM), the primary free water (LZFPM), and the supplemental free water (LZFWM). SAC-SMA’s processes and parameters are illustrated in more detail in Fig. 5.2. It is indicated in the figure that there are four principal forms of runoff generated by SAC-SMA: 1) direct runoff on the impervious area, 2) surface runoff when the upper zone free water storage is filled and the precipitation intensity is greater than percolation and interflow rate, 3) the lateral interflow from upper zone free water storage, and 4) primary baseflow. The direct runoff is composed of the impervious runoff over the permanent impervious area and the direct runoff on the temporal impervious...
Figure 5.2. Major SAC-SMA processes and their corresponding parameters. RIVA- Riparian vegetation area. SIDE-Ratio of deep recharge to channel base flow. RSERV- Fraction lower zone free water not transferable to tension water. The full description of other parameter names can be found in the table 5.2. The parameters in the shaded boxes pertain to storages or states.

area. The permanent impervious area, represented by parameter PCTIM (percent of impervious area), represents constant impervious areas such as pavements. The temporal impervious area, represented by parameter ADIMP (additional impervious area), includes the filling of small reservoirs, marshes, and temporal seepage outflow areas which become impervious when the upper zone tension water is filled. Prior work [192] has shown that thirteen out of sixteen parameters control model performance and must be calibrated. Feasible ranges of these thirteen parameters are presented by Boyle et al. [18] and also used in the calibration studies of Tang et al. [97, 125] and Vrugt et al. [53] (see Table 5.2). As shown in Table 5.2, the maximum allowable value of ADIMP specified by the author is 0.4 indicating that 40% of the watershed area is the additional impervious area, which can lead to large direct runoff under wet conditions.

5.4 Case study

5.4.1 Juniata watershed description

The Juniata Watershed, part of the Susquehanna River Basin, covers an area of 8800 km² in the ridge and valley region of the Appalachian Mountains of south
central Pennsylvania. The watershed is within the US NWS mid-Atlantic river forecast center (MARFC) area of forecast responsibility. The primary aquifer formations are composed of sedimentary and carbonate rocks that are presented in alternating layers of sandstone, shale, and limestone. Approximately, 67 percent of the watershed is forested, 23 percent is agricultural, 7 percent is developed area, and the rest is mine lands, water, or miscellaneous. There are 11 major sub-watersheds (see Fig. 5.3), among which, RTBP1, LWSP1, MPLP1, and NPTP1 have heavily controlled flows from reservoirs. Our preliminary analysis of the watershed focused on 7 headwater sub-watersheds where flows are not managed. Fig. 5.5 illustrates our preliminary analysis of the hydrologic conditions within the seven sub-watersheds by plotting their flow duration curves as well as monthly averages for streamflow, potential evaporation, and precipitation. Fig. 5.4 shows that the SPKP1 (Spruce Creek) and SXTP1 (Saxton) watersheds have distinctly different flow regimes. In the remainder of our study, we have evaluated the model sensitivities within these two watersheds using the four sensitivity analysis tools introduced in Section 5.2. As will be discussed in more detail in Section 5.5, our analysis evaluates model sensitivities for different temporal and spatial scales (i.e., SPKP1 and SXTP1 have drainage areas of 570 and 1960 km$^2$, respectively).
Figure 5.4. Hydrologic conditions of headwater sub-watersheds in the Juniata River basin.

5.4.2 Data set

The SAC-SMA/SNOW-17 lumped model used required input forcing data consisting of precipitation, potential evapotranspiration (PE), and air temperature. The precipitation data are next generation radar (NEXRAD) multisensor precipitation estimator data from the US NWS. Hourly data for precipitation and air temperature were available from January 1st, 2001 to December 31st, 2003. The observed streamflow in the same period was obtained from United States Geological Survey (USGS) gauge stations located at the outlets of the SPKP1 and SXTP1 watersheds.
5.5 Computational experiment

5.5.1 Model setup and parameterizations

In this study, we used a Linux computing cluster with 133 computer nodes composed of dual or quad AMD Opteron processors and 64 GB of RAM. Two month warmup periods (January 1<sup>st</sup> to February 28<sup>th</sup>, 2001) were used to reduce the influence of initial conditions. Model performance was evaluated using three different time intervals (1 hour, 6 hours, 24 hours) to test how parameter sensitivities change due to different prediction time scales. The <i>a priori</i> parameter settings used for the SNOW-17 and SAC-SMA models were based on the recommendations of the Mid-Atlantic River Forecasting Center of the US NWS.

The primary algorithmic parameters for PEST were set based on the recommendations of Doherty [86]. The initial Marquardt lambda and its adjust factor were set to be 5 and 2 respectively. When calculating the derivatives, the parameters were incremented by a fraction of the current parameters’ values subject to the absolute increment lower bounds. The fraction is 0.01 and the lower bounds vary from parameter to parameter based on their magnitudes. The parameter estimation process terminates if one of the following conditions is satisfied: 1) the number of iterations exceeds 30; 2) the relative difference between the objective value of the current iteration and the minimum objective value achieved to date is less than 0.01 for 3 successive iterations; 3) the algorithm fails to lower the objective value over 3 successive iterations; 4) the magnitude of the maximum relative parameter change between optimization iterations is less than 0.01 over 3 successive iterations.

Statistical sample sizes are key parameters for RSA, ANOVA, and Sobol’s method. In this study, the sample sizes were configured based on both literature recommendations and experiments by observing the convergence and reproducibility of the sensitivity analysis results. Sieber and Uhlenbrook [30] used a sample size of 10 times the number of perturbed parameters while doing sensitivity analysis on a distributed catchment model using LHS. However, the experimental analysis showed this is far from enough for our study. Examining statistical convergence as a function of increasing sample size, we determined a size of 10,000 was sufficient for LHS in RSA. For the ANOVA method, typically the $F$ values increase for
the sensitive parameters with increases in sample size [93]. Our analysis of convergence for the ANOVA method’s F-values and parameter sensitivity rankings showed that a sample size of 10,000 was sufficient when using IFFD sampling. For Sobol’s quasi-random sequence Sobol’ [184] states that additional uniformity can be obtained if the sample size is increased according to the function $n = 2^k$, where $k$ is an integer. Building on this recommendation, our analysis showed that Sobol’s sensitivity indices converged and were reproducible using a sample size 8,192 ($2^{13}$).

5.5.2 Objective functions

Two different model performance objective functions were used to screen the sensitivity of SAC-SMA and SNOW17 for high streamflow and low streamflow. The first objective was the non-transformed root mean square error (RMSE) objective, which is largely dominated by peak flow prediction errors due to the use of squared residuals. The second objective was formulated using a Box-Cox transformation of the hydrograph ($z = [(y+1)^\lambda - 1]/\lambda$ where $\lambda = 0.3$) as recommended by Misirli et al. [114] to reduce the impacts of heteroscedasticity in the RMSE calculations (also increasing the influence of low flow periods).

5.5.3 Bootstrap confidence intervals

For ANOVA and Sobol’s method, the $F$ values and sensitivity indices can have a high degree of uncertainty due to random number generation effects [96, 193]. In this study, we used the bootstrap method [194] to provide confidence intervals for the parameter sensitivity rankings for both ANOVA and Sobol’s method. Essentially, the samples generated by IFFD or Sobol’s sequence were resampled $N$ times when calculating the $F$ values or sensitivity indices for each parameter, resulting in a distribution of the $F$ values or indices. The moment method [193] was adopted for acquiring the bootstrap confidence intervals (BCIs) for this chapter. The moment method is based on large sample theory and requires a sufficiently large resampling dimension to yield symmetric 95% confidence intervals. In this study, the resample dimension $N$ was set to 2,000 based on prior literature discussions as well as computational experiments that confirmed a symmetric distribution for standard errors. Readers interested in detailed descriptions of the bootstrapping
method used in this chapter can reference the following sources [193, 194].

5.5.4 Evaluation of sensitivity analysis results

As argued by Andres [181], good sensitivity analysis tools should generate repeatable results using a different sample set to evaluate model sensitivities. The effectiveness of a sensitivity analysis method refers to its ability to correctly identify the influential parameters controlling a model’s performance. Building on Andres [181], we have tested the effectiveness of each of the sensitivity methods using an independent LHS-based random draw of 1000 parameter groups for the 18 parameters analyzed in this study.

The independent sample and the sensitivity classifications from each of the sensitivity analysis methods were combined to develop three parameter sets. Set 1 consists of the full randomly generated independent sample set of 1000 parameter groups. In Set 2, the parameters classified as highly sensitive or sensitive are set to a priori values while the remaining insensitive parameters are allowed to vary randomly. Lastly, in Set 3 the parameters classified as being highly sensitive or sensitive vary randomly and the insensitive parameters are set to a priori values.

Varying parameters that are correctly classified as insensitive in Set 2 should theoretically yield a zero correlation with the full random sample of Set 1 (i.e., plot as a horizontal line). If some parameters are incorrectly classified as insensitive then the scatter plots show deviations from a horizontal line and increased correlation coefficients. Conversely, if the correct subset of sensitive parameters is sampled randomly (i.e., Set 3) then they should be sufficient to capture model output from the random samples of the full parameter set in Set 1 yielding a linear trend with an ideal correlation coefficient of 1. We extended the evaluation methodology of Andres [181] by calculating the corresponding correlation coefficients instead of using scatter plots.

5.6 Results

Sections 5.6.1-5.6.4 present the results attained for each of the four sensitivity analysis methods tested in this chapter. Results are presented for the SPKP1
and SXTP1 watersheds at 1 hour, 6 hour, and 24 hour timescales. Section 5.6.5 then provides a detailed analysis of how the results from each sensitivity method compare in terms of their selection of highly sensitive, sensitive, and insensitive parameters for the SAC-SMA/SNOW-17 lumped model. Additionally, Section 5.6.5 builds on the work of Andres [181] to evaluate the relative effectiveness of the methods in identifying the key input parameters controlling model performance. Detailed conclusions on how individual watershed properties impact model performance are beyond the scope of this chapter.

Before discussing the sensitivity results in detail, it is worth noting that there are several ways that sensitivity analysis methods can be evaluated and used in the context of watershed model identification and evaluation. The current study builds on the optimization research of Tang et al. [97] by focusing on how well PEST, RSA, ANOVA, and Sobol’s method can identify the set of model input parameters that control model performance. Successful screening of the relative importance of input parameters and their interactions can help to limit the dimensionality of calibration search problems and serve to enhance the efficiency of uncertainty analysis. Recall from Section 5.5 that the model performance objectives used in this study evaluate the influence of high streamflow conditions via the RMSE measure and low streamflow conditions via the Box-Cox transformed RMSE (TRMSE). Small values of these measures implies that the SAC-SMA/SNOW-17 streamflow projections closely match observations in the simulated period.

### 5.6.1 Sensitivity results for PEST

In the case of PEST, sensitivities are computed using the Jacobean derivative-based composite measures defined in equation 5.1. The method is termed local because the composite derivatives are evaluated at a single point in the parameter space deemed locally optimal by the Gauss-Marquardt-Levenberg algorithm. Tables 5.3 and 5.4 provide the sensitivities computed by PEST for the RMSE and TRMSE objectives, respectively. In the tables, highly sensitive parameters are designated with dark grey shading, sensitive parameters have light grey shading, and insensitive parameters are not shaded. The SNOW-17 and SAC-SMA parameters are listed separately as are the 1 hour, 6 hour, and 24 hour results for each
Table 5.3. PEST sensitivities based on the RMSE measure. Dark gray shading designates highly sensitive parameters defined using a threshold value of 1.0. Light gray designates sensitive parameters defined using a threshold value of 0.01. White cells in the table designate insensitive parameters.

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter</th>
<th>SPKPI 1h</th>
<th>SPKPI 6h</th>
<th>SPKPI 24h</th>
<th>SXTP1 1h</th>
<th>SXTP1 6h</th>
<th>SXTP1 24h</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOW-17</td>
<td>SCF</td>
<td>0.04</td>
<td>0.18</td>
<td>0.31</td>
<td>0.11</td>
<td>0.35</td>
<td>0.55</td>
</tr>
<tr>
<td></td>
<td>MFMAX</td>
<td>0</td>
<td>0.07</td>
<td>0.01</td>
<td>0</td>
<td>0.05</td>
<td>0.11</td>
</tr>
<tr>
<td></td>
<td>MFMIN</td>
<td>0.03</td>
<td>0.16</td>
<td>0.42</td>
<td>0.03</td>
<td>0.18</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td>UADJ</td>
<td>0.01</td>
<td>0.10</td>
<td>0.50</td>
<td>0.02</td>
<td>0.13</td>
<td>0.35</td>
</tr>
<tr>
<td></td>
<td>SI</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SNOW-17 MFMIN</td>
<td>0.03</td>
<td>0.16</td>
<td>0.42</td>
<td>0.03</td>
<td>0.18</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td>UADJ</td>
<td>0.01</td>
<td>0.10</td>
<td>0.50</td>
<td>0.02</td>
<td>0.13</td>
<td>0.35</td>
</tr>
<tr>
<td></td>
<td>SI</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>UZTWM</td>
<td>0</td>
<td>0</td>
<td>0.01</td>
<td>0</td>
<td>0</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>UZFWM</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>UZK</td>
<td>0.03</td>
<td>0.07</td>
<td>0.16</td>
<td>0.55</td>
<td>1.15</td>
<td>2.26</td>
</tr>
<tr>
<td></td>
<td>PCTIM</td>
<td>0.30</td>
<td>0.57</td>
<td>1.03</td>
<td>0.69</td>
<td>1.65</td>
<td>3.05</td>
</tr>
<tr>
<td></td>
<td>ADIMP</td>
<td>0.17</td>
<td>0.34</td>
<td>0.60</td>
<td>0.36</td>
<td>0.79</td>
<td>1.71</td>
</tr>
<tr>
<td></td>
<td>ZPerc</td>
<td>0.01</td>
<td>0.01</td>
<td>0.07</td>
<td>0</td>
<td>0.01</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td>LEXP</td>
<td>0.01</td>
<td>0.01</td>
<td>0.07</td>
<td>0</td>
<td>0.01</td>
<td>0.04</td>
</tr>
<tr>
<td></td>
<td>LZTWM</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LZFSM</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LZFPM</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LZSK</td>
<td>0.14</td>
<td>0.20</td>
<td>0.58</td>
<td>1.30</td>
<td>5.04</td>
<td>10.06</td>
</tr>
<tr>
<td></td>
<td>LZPK</td>
<td>1.59</td>
<td>1.11</td>
<td>5.54</td>
<td>0.21</td>
<td>0.14</td>
<td>32.61</td>
</tr>
<tr>
<td></td>
<td>PFREE</td>
<td>0.01</td>
<td>0.01</td>
<td>0.08</td>
<td>0.03</td>
<td>0.10</td>
<td>0.03</td>
</tr>
</tbody>
</table>

As a caveat, the thresholds used to differentiate highly sensitive, sensitive, and insensitive parameters are based only on the relative magnitudes of the derivatives given in each column, making them subjective and somewhat arbitrary. The thresholds were determined by ranking each column in ascending order and then plotting the relative magnitudes of the derivatives. Results were classified as either highly sensitive or sensitive where the derivative values changed the most significantly. Insensitive parameters had small derivative values that could not be distinguished. Note different thresholds were used for Tables 5.3 and 5.4 since the Box-Cox transformation reduced the original range of RMSE by approximately an order of magnitude. The results in Tables 5.3 and 5.4 show that PEST did not detect significant changes in parameter sensitivities for high flow (RMSE) versus low flow (TRMSE) conditions. Also differences in the times-scales of predictions as well as watershed locations did not significantly change the PEST sensitivity designations in both tables. Overall PEST found the parameters for impervious cover (PCTIM, ADIMP) and those for storage depletion rates (UZK, LZPK, LZSK) significantly impacted model performance, especially for daily time-scale predictions. The mean water-equivalent threshold for snow cover (SI), upper zone storage
Table 5.4. PEST sensitivities based on the TRMSE measure. Dark gray shading designates highly sensitive parameters defined using a threshold value of 0.1. Light gray designates sensitive parameters defined using a threshold value of 0.001. White cells in the table designate insensitive parameters.

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter</th>
<th>SPKP1 1h</th>
<th>SPKP1 6h</th>
<th>SPKP1 24h</th>
<th>SXTP1 1h</th>
<th>SXTP1 6h</th>
<th>SXTP1 24h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0.004</td>
<td>0.014</td>
<td>0.022</td>
<td>0.007</td>
<td>0.020</td>
<td>0.033</td>
</tr>
<tr>
<td></td>
<td>SCF</td>
<td>0.001</td>
<td>0.006</td>
<td>0.001</td>
<td>0.002</td>
<td>0.001</td>
<td></td>
</tr>
<tr>
<td>SNOW-17</td>
<td>MFMAX</td>
<td>0.002</td>
<td>0.008</td>
<td>0.019</td>
<td>0.008</td>
<td>0.016</td>
<td>0.025</td>
</tr>
<tr>
<td></td>
<td>MFMIN</td>
<td>0.001</td>
<td>0.005</td>
<td>0.009</td>
<td>0.001</td>
<td>0.007</td>
<td>0.010</td>
</tr>
<tr>
<td></td>
<td>UADJ</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>SI</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SAC-SMA</td>
<td>UZTWM</td>
<td>0</td>
<td>0</td>
<td>0.001</td>
<td>0</td>
<td>0</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>UZFWM</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>UZK</td>
<td>0.007</td>
<td>0.008</td>
<td>0.047</td>
<td>0.061</td>
<td>0.095</td>
<td>0.134</td>
</tr>
<tr>
<td></td>
<td>PCTIM</td>
<td>0.041</td>
<td>0.080</td>
<td>0.159</td>
<td>0.100</td>
<td>0.227</td>
<td>0.419</td>
</tr>
<tr>
<td></td>
<td>ADIMP</td>
<td>0.020</td>
<td>0.038</td>
<td>0.078</td>
<td>0.020</td>
<td>0.045</td>
<td>0.121</td>
</tr>
<tr>
<td></td>
<td>ZPERC</td>
<td>0</td>
<td>0</td>
<td>0.003</td>
<td>0.001</td>
<td>0.002</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>REXP</td>
<td>0.001</td>
<td>0.001</td>
<td>0.001</td>
<td>0</td>
<td>0.001</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td>LZTWM</td>
<td>0</td>
<td>0</td>
<td>0.001</td>
<td>0</td>
<td>0</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td>LZFSM</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>LZFPM</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>LZSK</td>
<td>0.014</td>
<td>0.024</td>
<td>0.075</td>
<td>0.059</td>
<td>0.353</td>
<td>0.006</td>
</tr>
<tr>
<td></td>
<td>LZPK</td>
<td>0.258</td>
<td>0.449</td>
<td>0.713</td>
<td>0.332</td>
<td>5.024</td>
<td>2.513</td>
</tr>
<tr>
<td></td>
<td>PFREE</td>
<td>0.004</td>
<td>0.007</td>
<td>0.016</td>
<td>0.005</td>
<td>0.006</td>
<td>0.009</td>
</tr>
</tbody>
</table>

parameters (UZTWM, UZFWM), and lower zone storage parameters (LZTWM, LZFSM, and LZFPM) were classified by PEST as being the least sensitive.

5.6.2 RSA Results

As described in Section 5.2.2.2, a visual extension of RSA [24, 25, 80, 87] was used to evaluate parameter sensitivities for the SAC-SMA/SNOW-17 lumped model. Results were computed for the same timescales and watersheds as were presented for PEST. Given the large number of results analyzed, Figures 5.5 and 5.6 provide sample plots for our RSA analysis, whereas the full sensitivity classifications are summarized in Tables 5.5 and 5.6. In Figs. 5.5 and 5.6 each model parameter has its own plot with its range on the horizontal axis and its cumulative normalized RMSE distribution value on the vertical axis. In the plots, color shading is used to differentiate the likelihoods of each one of the ten bins used to divide the input parameter samples. High likelihood bins plotted in purple represent portions of the parameters’ ranges where low RMSE values are expected. In the context of sensitivity analysis, RSA measures the distribution of model responses that result from the 10,000 Latin hypercube input parameter groups sampled. When parameters are insensitive (see the SNOW-17 results shown in Fig. 5.5) each of the 10
Figure 5.5. RSA (Regional Sensitivity Analysis) plot for Snow17 parameters in the SPKP1 watershed. The objective function is RMSE based on a 1-hour time interval.

Table 5.5. RSA sensitivities based on the RMSE measure. Dark gray shading designates highly sensitive (HS) parameters. Light gray designates sensitive (S) parameters. White cells in the table designate parameters that are not sensitive (NS).

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter</th>
<th>SPKP1 1h</th>
<th>SPKP1 6h</th>
<th>SPKP1 24h</th>
<th>SXTP1 1h</th>
<th>SXTP1 6h</th>
<th>SXTP1 24h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1h</td>
<td>6h</td>
<td>24h</td>
<td>1h</td>
<td>6h</td>
<td>24h</td>
</tr>
<tr>
<td></td>
<td>SCF</td>
<td>S</td>
<td>S</td>
<td>NS</td>
<td>S</td>
<td>S</td>
<td>NS</td>
</tr>
<tr>
<td>SNOX-17</td>
<td>MFMAX</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td></td>
<td>MFMIN</td>
<td>NS</td>
<td>NS</td>
<td>S</td>
<td>NS</td>
<td>S</td>
<td>NS</td>
</tr>
<tr>
<td></td>
<td>UADJ</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td></td>
<td>SI</td>
<td>NS</td>
<td>NS</td>
<td>S</td>
<td>NS</td>
<td>S</td>
<td>NS</td>
</tr>
<tr>
<td></td>
<td>UZTWM</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>NS</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>UZFWM</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>UZK</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td></td>
<td>PCTIM</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>ADIMP</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>ZPERC</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td></td>
<td>REXP</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>SAC-SMA</td>
<td>LZTWM</td>
<td>HS</td>
<td>HS</td>
<td>HS</td>
<td>HS</td>
<td>HS</td>
<td>HS</td>
</tr>
<tr>
<td></td>
<td>LZFWM</td>
<td>S</td>
<td>NS</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>LZFPM</td>
<td>NS</td>
<td>NS</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>NS</td>
</tr>
<tr>
<td></td>
<td>LZSK</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>HS</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>LZFPK</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>PFREE</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
</tbody>
</table>

sample bins plot over each other in linear trend lines that are representative of uniformly distributed RMSE values. Sensitive parameters produced highly dispersed bin lines such as those shown for LZTWM shown in Fig. 5.6.

The SAC-SMA/SNOW-17 sensitivity classifications resulting from RSA are presented in Tables 5.5 and 5.6. The classifications represent our qualitative interpretation of visual plots similar to those in Figs. 5 and 6 for each timescale and each watershed. As is standard in hydrologic applications of RSA [e.g., [25, 80]], only individual parameter impacts on model performance are considered and parameter interactions have been neglected. Analysis of Tables 5.5 and 5.6 show
Figure 5.6. RSA (Regional Sensitivity Analysis) plot for SAC-SMA parameters in the SPKP1 watershed. The objective function is RMSE based on a 1-hour time interval.

Table 5.6. RSA sensitivities based on the TRMSE measure. Dark gray shading designates highly sensitive (HS) parameters. Light gray designates sensitive (S) parameters. White cells in the table designate parameters that are not sensitive (NS).

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter</th>
<th>SPKP1 1h</th>
<th>SPKP1 6h</th>
<th>SPKP1 24h</th>
<th>SXTPI 1h</th>
<th>SXTPI 6h</th>
<th>SXTPI 24h</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCF</td>
<td>S</td>
<td>NS</td>
<td>S</td>
<td>S</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>MFMAX</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>MFMIN</td>
<td>NS</td>
<td>S</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>S</td>
</tr>
<tr>
<td>UADJ</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>SI</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>UZTWM</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>UZFWM</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>UZK</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>PCTIM</td>
<td>NS</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>ADIMP</td>
<td>HS</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>NS</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>ZPERC</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>REXP</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
</tr>
<tr>
<td>LZTWM</td>
<td>HS</td>
<td>HS</td>
<td>HS</td>
<td>HS</td>
<td>HS</td>
<td>HS</td>
<td>HS</td>
</tr>
<tr>
<td>LZFSM</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>LZFPM</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>LZSK</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>LZPK</td>
<td>NS</td>
<td>NS</td>
<td>NS</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>PFREE</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
</tbody>
</table>
changes in sensitivity when comparing across timescales, watersheds, and model performance objectives. Examples of these changes include the increased importance of the SNOW-17 parameters such as the mean water-equivalent above which 100-percent cover exists (SI) for the RMSE measure (i.e., high flow) and the minimum melt factor for non-rain periods (MFMIN) for the TRMSE measure (i.e., low flow) at the daily timescale. Both the RMSE measure and the TRMSE measure identified the vadose zone storage (LZTWM) as the most sensitive parameter in all of the tested cases. Shifting the focus from high flow to low flow using the TRMSE measure resulted in the percolation factor (PFREE) and lower zone free water primary maximum storage (LZFPM) being classified as being sensitive.

5.6.3 Sensitivity results for ANOVA

Recall that ANOVA is a parametric analysis of variance that uses the assumption of normally distributed model responses (RMSE and TRMSE for streamflow in this study) to partition variance contributions between single parameters and parameter interactions. In this study, a second order ANOVA model (i.e., a model that considers pair wise parameter interactions) was fit to the model outputs and the F-test is used to evaluate the statistical significance of each parameter’s or parameter interaction’s impact on the model output. Higher F-values indicate higher significance or sensitivity. Additionally, the coefficient of determination $r^2$ can be used to measure if incorporating parameter interactions into the ANOVA model improves its ability to represent model output variability [93]. Because random sampling can introduce significant uncertainty into the calculation of F-values, we have followed the recommendations of Archer et al. [193] and used statistical bootstrapping to provide 95% confidence intervals for our ANOVA sensitivity rankings. Tables 5.7 and 5.8 provide F-values for each parameter as well as its bootstrapped confidence interval. Tabular presentation of the ANOVA results improved their clarity since the F-values ranged over 4 orders of magnitude [0.25 - 4000] making plots difficult to interpret.

Tables 5.7 and 5.8 are formatted similarly to the prior sensitivity tables where highly sensitive parameters have dark grey shading, sensitive parameters have light grey shading, and insensitive parameters have no shading. These classifications
Table 5.7. ANOVA single parameter sensitivities based on the RMSE measure. Dark gray shading designates highly sensitive parameters defined using a threshold F value of 460. Light gray designates sensitive parameters defined using a threshold F value of 4.6. White cells in the table designate insensitive parameters. The values in the brackets provide the 95% confidence interval for the F-values (i.e., the unbracketed value ± the bracketed value yields the confidence interval).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>1h</th>
<th>6h</th>
<th>24h</th>
<th>1h</th>
<th>6h</th>
<th>24h</th>
</tr>
</thead>
<tbody>
<tr>
<td>SCP</td>
<td>57.02 [21.78]</td>
<td>94.63 [27.57]</td>
<td>0.24 [2.40]</td>
<td>80.38 [25.71]</td>
<td>71.50 [23.80]</td>
<td>0.65 [2.79]</td>
</tr>
</tbody>
</table>

Table 5.8. ANOVA single parameter sensitivities based on the TRMSE measure. Dark gray shading designates highly sensitive parameters defined using a threshold F value of 460. Light gray designates sensitive parameters defined using a threshold F value of 4.6. White cells in the table designate insensitive parameters. The values in the brackets provide the 95% confidence interval for the F-values (i.e., the unbracketed value ± the bracketed value yields the confidence interval).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>1h</th>
<th>6h</th>
<th>24h</th>
<th>1h</th>
<th>6h</th>
<th>24h</th>
</tr>
</thead>
<tbody>
<tr>
<td>SH</td>
<td>1.07 [3.45]</td>
<td>0.82 [2.20]</td>
<td>1.39 [3.85]</td>
<td>0.60 [2.95]</td>
<td>0.81 [6.78]</td>
<td>3.72 [5.69]</td>
</tr>
</tbody>
</table>

Table 5.7, Table 5.8, and the text that follows were based on the F-distribution where a threshold of 4.6 represents less than a 1-percent chance of misclassifying a parameter as sensitive. As can be seen in the tables, some parameters’ F-values were up to three orders of magnitude larger than 4.6. A threshold of 460 was used to classify parameters as being highly sensitive. Although the threshold used to classify highly sensitive parameters is subjective, it accurately captures those parameters with very large F-values.
Table 5.9. Coefficients of determination for the ANOVA model. R1 designates a 1\textsuperscript{st} order ANOVA model that neglects parameter interactions. R2 designates a 2\textsuperscript{nd} order ANOVA model that accounts for pairwise parameter interactions.

<table>
<thead>
<tr>
<th>Order</th>
<th>RMSE</th>
<th>TRMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SPKP1</td>
<td>SXTP1</td>
</tr>
<tr>
<td>1h</td>
<td>0.535</td>
<td>0.755</td>
</tr>
<tr>
<td>6h</td>
<td>0.730</td>
<td>0.861</td>
</tr>
</tbody>
</table>

Analysis of Table 5.7 shows that for the high-flow RMSE objective, the most significant differences in sensitivities across timescales and across watersheds involved SNOW-17 parameters. The results show increasing sensitivities for the minimum melt factor for non-rain periods (MFMIN) at the 6-hour and daily timescales. Overall, Table 5.7 shows that most of the SAC-SMA parameters are sensitive for high flow conditions regardless of timescale or watershed. The high flow RMSE analysis identified the lower zone storage (LZTWM) as having the highest influence on model variance while the upper zone free water lateral depletion rate (UZK) is rated to have the least impact.

In Table 5.8 the ANOVA results using the low flow TRMSE objective are substantially different from those for high flow in Table 5.7. For low flow conditions, fewer parameters are classified as being sensitive. Table 5.8 shows a general reduction relative to Table 5.7 in the influence of the upper zone free water storage (UZFWM) and an increase in the importance of the upper zone tension water storage (UZTWM) as well as the percolation factor (PFREE).

Beyond single parameter sensitivities, the coefficients of determination in Table 5.9 show that 2\textsuperscript{nd} order interactions (or pairwise parameter interactions) improve the accuracy of the ANOVA model, which means the model better represents the total variance of the SAC-SMA/SNOW-17 model output. The coefficients of determination show that 2\textsuperscript{nd} order parameter interactions improve the ANOVA models’ performances by up to 40%. Figure 5.7 illustrates the 2\textsuperscript{nd} order parameter interactions impacting the SAC-SMA/SNOW-17 model. Second order analysis changes the degrees of freedom used when analyzing the F-distribution making it necessary to define a new threshold in Fig. 5.7. An F-value threshold of 3.32 designates at least a 99% likelihood of being sensitive. Again higher F-values imply higher sensitivity.

Figure 5.7 provides a more detailed portrayal of how parameter sensitivities...
Figure 5.7. (a) ANOVA second order parameter interactions based on the RMSE measure. (b) ANOVA second order parameter interactions based on the TRMSE measure. Circles represent statistically significant F-values defined using the threshold value of 3.32. The color legends and shading represent the F-value magnitudes and ranges.
change across timescales for each of the watershed models. The RMSE results in Fig. 5.7b show that interactively sensitive parameters varied across watersheds as well as timescales. The results in Fig. 5.7b and Table 5.9 show that the SXTP1 watershed possesses more TRMSE-based ANOVA interactions than the SPKP1 watershed. The results imply each watershed model has a "unique" set of parameter interactions impacting its performance [195].

5.6.4 Sensitivity results for Sobol’s method

Recall from Section 5.2.2.4 that Sobol’s method decomposes the overall variance of the sampled SAC-SMA/SNOW-17 model output to compute 1st order (single parameter), 2nd order (two parameter), and total order sensitivity indices. These indices are presented as percentages and have straightforward interpretations as representing the percent of total model output variance contributed by a given parameter or parameter interaction. The total order indices are the most comprehensive measures of a single parameter’s sensitivity since they represent the summation of all variance contributions involving that parameter (i.e., its 1st order contribution plus all of its pairwise interactions).

Table 5.10 shows the relative importance of 1st and 2nd order effects for all of the cases analyzed. Readers should note that the truncation and Monte Carlo approximations of the integrals required in Sobol’s method can lead to small numerical errors [e.g., see [95, 96, 193]] such as slightly negative indices or for example in Table 5.10 the few cases where 1st and 2nd order effects sum to be slightly larger than 1. In this study these effects were very small and did not impact parameter rankings. Table 5.10 supports our analysis assumption that 1st and 2nd order parameter sensitivities explain nearly all of the variance in the SAC-SMA/SNOW-17 model’s output distributions. The table also shows that the importance of 2-parameter interactions ranged from 3% to 40% of the total variance depending on the model performance objective, the prediction timescale, and the watershed. Except for SXTP1 6-hour test case, the results indicate that there were more parameter interactions for the RMSE measure compared to the TRMSE measure.

Tables 5.11 and 5.12 summarize the total order indices (i.e., total variance
Table 5.10. Summations of Sobol's sensitivity indices for 1st order and 2nd order contributions to model output variance.

<table>
<thead>
<tr>
<th>Order</th>
<th>RMSE</th>
<th>TRMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SPKP1</td>
<td>SXTP1</td>
</tr>
<tr>
<td>1st</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2nd</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.11. Total order sensitivity indices from Sobol's method computed using the RMSE measure. Dark gray shading designates highly sensitive parameters defined using a threshold value of 0.1. Light gray designates sensitive parameters defined using a threshold value of 0.01. White cells in the table designate insensitive parameters. The values in the brackets provide the 95% confidence interval for the indices' values (i.e., the unbracketed value ± the bracketed value yields the confidence interval).

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter</th>
<th>SPKP1</th>
<th>SXTP1</th>
<th>SPKP1</th>
<th>SXTP1</th>
<th>SPKP1</th>
<th>SXTP1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1h</td>
<td>6h</td>
<td>24h</td>
<td>1h</td>
<td>6h</td>
<td>24h</td>
</tr>
<tr>
<td>SNOW-17</td>
<td>SCF</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00*</td>
<td>0.00*</td>
<td>0.00*</td>
<td>0.00*</td>
</tr>
<tr>
<td></td>
<td>MFMAX</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00*</td>
<td>0.00*</td>
<td>0.00*</td>
<td>0.00*</td>
</tr>
<tr>
<td></td>
<td>MFMIN</td>
<td>0.00*</td>
<td>0.00*</td>
<td>0.00*</td>
<td>0.00*</td>
<td>0.00*</td>
<td>0.00*</td>
</tr>
<tr>
<td></td>
<td>UADJ</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00*</td>
<td>0.00*</td>
<td>0.00*</td>
<td>0.00*</td>
</tr>
<tr>
<td></td>
<td>SI</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00*</td>
<td>0.00*</td>
<td>0.00*</td>
<td>0.00*</td>
</tr>
<tr>
<td></td>
<td>UZTWM</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td></td>
<td>UZFWM</td>
<td>0.20</td>
<td>0.20</td>
<td>0.16</td>
<td>0.16</td>
<td>0.16</td>
<td>0.16</td>
</tr>
<tr>
<td></td>
<td>UZK</td>
<td>0.01</td>
<td>0.01</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td></td>
<td>PCTIM</td>
<td>0.11</td>
<td>0.11</td>
<td>0.08</td>
<td>0.08</td>
<td>0.08</td>
<td>0.08</td>
</tr>
<tr>
<td></td>
<td>ADIMP</td>
<td>0.30</td>
<td>0.30</td>
<td>0.13</td>
<td>0.13</td>
<td>0.13</td>
<td>0.13</td>
</tr>
<tr>
<td></td>
<td>ZPERC</td>
<td>0.03</td>
<td>0.03</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
<td>0.05</td>
</tr>
<tr>
<td>SAC-SMA</td>
<td>REXP</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
<td>0.01</td>
</tr>
<tr>
<td></td>
<td>LZTWM</td>
<td>0.48</td>
<td>0.48</td>
<td>0.48</td>
<td>0.48</td>
<td>0.48</td>
<td>0.48</td>
</tr>
<tr>
<td></td>
<td>LZFSM</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td></td>
<td>LZFPM</td>
<td>0.03</td>
<td>0.03</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
</tr>
<tr>
<td></td>
<td>LZSK</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
<td>0.06</td>
</tr>
<tr>
<td></td>
<td>LZPK</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
<td>0.03</td>
</tr>
<tr>
<td></td>
<td>PFREE</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
<td>0.02</td>
</tr>
</tbody>
</table>

* Negative mean value was set to zero

Contributions) for the SAC-SMA/SNOW-17 parameters analyzed. Again highly sensitive parameters are designated with dark grey shading, sensitive parameters have light grey shading, and insensitive parameters are not shaded. In all of the results presented for Sobol's method, parameters classified as highly sensitive had to contribute on average at least 10-percent of the overall model variance and sensitive parameters had to contribute at least 1-percent. These thresholds are subjective and their ease-of-satisfaction decreases with increasing numbers of parameters or parameter interactions. In Tables 5.11 and 5.12 the total order indices again show that the model performance objective, the prediction timescale, and the watershed all heavily impact the SAC-SMA/SNOW-17 sensitivities.

In both tables, the SNOW-17 parameters contributed minimally to the over-
Table 5.12. Total order sensitivity indices from Sobol’s method computed using the TRMSE measure. Dark gray shading designates highly sensitive parameters defined using a threshold value of 0.1. Light gray designates sensitive parameters defined using a threshold value of 0.01. White cells in the table designate insensitive parameters. The values in the brackets provide the 95% confidence interval for the indices’ values (i.e., the unbracketed value ± the bracketed value yields the confidence interval).

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter</th>
<th>SPKP1 1h</th>
<th>6h</th>
<th>24h</th>
<th>SXTP1 1h</th>
<th>6h</th>
<th>24h</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0.01 [0.00]</td>
<td>0.01 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.01 [0.00]</td>
<td>0.01 [0.00]</td>
<td>0.00 [0.00]</td>
</tr>
<tr>
<td></td>
<td>SCF</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
</tr>
<tr>
<td>SNOW-17</td>
<td>MFMAX</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
</tr>
<tr>
<td></td>
<td>MFMIN</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.01 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.01 [0.00]</td>
</tr>
<tr>
<td></td>
<td>UADJ</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
</tr>
<tr>
<td></td>
<td>SI</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
</tr>
<tr>
<td></td>
<td>UZTWM</td>
<td>0.14 [0.01]</td>
<td>0.14 [0.01]</td>
<td>0.10 [0.01]</td>
<td>0.14 [0.01]</td>
<td>0.13 [0.01]</td>
<td>0.08 [0.01]</td>
</tr>
<tr>
<td></td>
<td>UZFWM</td>
<td>0.02 [0.01]</td>
<td>0.03 [0.01]</td>
<td>0.05 [0.01]</td>
<td>0.02 [0.01]</td>
<td>0.07 [0.02]</td>
<td>0.10 [0.02]</td>
</tr>
<tr>
<td></td>
<td>UZK</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
</tr>
<tr>
<td></td>
<td>PCTIM</td>
<td>0.02 [0.00]</td>
<td>0.06 [0.01]</td>
<td>0.04 [0.01]</td>
<td>0.05 [0.01]</td>
<td>0.05 [0.01]</td>
<td>0.04 [0.01]</td>
</tr>
<tr>
<td></td>
<td>ADIMP</td>
<td>0.22 [0.01]</td>
<td>0.05 [0.01]</td>
<td>0.06 [0.01]</td>
<td>0.08 [0.01]</td>
<td>0.05 [0.01]</td>
<td>0.09 [0.01]</td>
</tr>
<tr>
<td></td>
<td>ZPERC</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.01 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
</tr>
<tr>
<td>SAC-SMA</td>
<td>REXP</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
</tr>
<tr>
<td></td>
<td>LZTWM</td>
<td>0.43 [0.02]</td>
<td>0.49 [0.02]</td>
<td>0.51 [0.02]</td>
<td>0.51 [0.02]</td>
<td>0.51 [0.02]</td>
<td>0.46 [0.02]</td>
</tr>
<tr>
<td></td>
<td>LZFSM</td>
<td>0.02 [0.01]</td>
<td>0.02 [0.00]</td>
<td>0.02 [0.00]</td>
<td>0.03 [0.01]</td>
<td>0.04 [0.01]</td>
<td>0.02 [0.01]</td>
</tr>
<tr>
<td></td>
<td>LZFPM</td>
<td>0.10 [0.01]</td>
<td>0.08 [0.01]</td>
<td>0.08 [0.01]</td>
<td>0.12 [0.01]</td>
<td>0.11 [0.01]</td>
<td>0.13 [0.01]</td>
</tr>
<tr>
<td></td>
<td>LZSK</td>
<td>0.02 [0.00]</td>
<td>0.02 [0.00]</td>
<td>0.02 [0.00]</td>
<td>0.05 [0.01]</td>
<td>0.06 [0.01]</td>
<td>0.04 [0.01]</td>
</tr>
<tr>
<td></td>
<td>LZPK</td>
<td>0.04 [0.01]</td>
<td>0.03 [0.01]</td>
<td>0.03 [0.01]</td>
<td>0.09 [0.01]</td>
<td>0.08 [0.01]</td>
<td>0.05 [0.01]</td>
</tr>
<tr>
<td></td>
<td>PFREE</td>
<td>0.13 [0.01]</td>
<td>0.15 [0.01]</td>
<td>0.16 [0.01]</td>
<td>0.23 [0.02]</td>
<td>0.23 [0.03]</td>
<td>0.27 [0.02]</td>
</tr>
</tbody>
</table>

* Negative mean value was set to zero

The variance of the simulation model’s output. Only the minimum melt factor for non-rain periods (MFMIN) parameter has a statistically significant sensitivity when the bootstrapped confidence intervals are considered. Tables 5.11 and 5.12 also insinuate that most of the SAC-SMA model parameters are sensitive. For the high-flow RMSE results, the lower zone tension water storage (LZTWM) and the additional impervious area (ADIMP) were the most sensitive SAC-SMA parameters. The upper zone storage parameters (UZTWM, UZFWM) and all of the lower zone parameters dominate model response for the low-flow TRMSE measure. In particular, the lower zone tension water storage (LZTWM) appears to be the dominant overall parameter as it explains about 50% of the output’s variance for each test case. Similar to ANOVA’s results, there are fewer parameters classified as being sensitive for the TRMSE measure versus RMSE.

Figure 5.8 provides a more detailed understanding of the total order indices presented in Tables 5.11 and 5.12. Similar to the ANOVA interaction plots in Section 5.6.3, these figures show the matrix of parameter interactions where circles designate pairings that contribute at least 1% of the overall model output variance.
The actual 2\textsuperscript{nd} order indices’ values are shown with the color shading defined in the plots’ legends. These plots show how the dominant parameters for both the RMSE and TRMSE measures tend to have the greatest number of interactions (e.g., LZTWM and PFREE in Fig. 5.7). Interestingly, there are very distinct differences for the parameter interactions for the two watersheds. When comparing the RMSE results in Fig. 8a with TRMSE results in Fig. 8b the shift from high-flow to low flow analysis tends to substantially decrease the importance of parameter interactions for the SPKP1 watershed, whereas no significant reduction was found for the SXTP1 watershed. Readers should note that our 1\% threshold for Sobol’s method is particularly conservative when analyzing Figs. 5.7a and 5.7b since the number of variables analyzed increases from 18 for 1\textsuperscript{st} order analysis to 162 parameter interactions in 2\textsuperscript{nd} order analysis.

### 5.6.5 Comparative summary of sensitivity methods

Sections 5.6.1–5.6.4 present classifications of SAC-SMA/SNOW-17 model parameters into three categories: (1) highly sensitive, (2) sensitive, and (3) insensitive. Given the large number of cases analyzed in this study, Fig. 5.9 was developed to provide a comparative summary of the results attained from the four sensitivity analysis methods. These figures show that there are distinct similarities and differences between the sensitivity classifications attained using each method. For example, despite the subjective decisions required to differentiate highly sensitive and sensitive parameters, generally RSA, ANOVA, and Sobol’s method agree on their classifications of the most sensitive parameters for each scenario.

All three global sensitivity methods (RSA, ANOVA, and Sobol’s method) show that the SAC-SMA/SNOW-17 model’s responses are “uniquely” determined by the performance objective specified, prediction timescale, and specific watershed being modeled [195]. Differences between the four sensitivity methods’ classifications as illustrated in Fig. 5.9 are particularly pronounced for parameters at the threshold between sensitive and insensitive. Significant differences exist between the classifications of PEST and those of the other three algorithms. One of the biggest discrepancies shown in the plots is that PEST generally found the SNOW-17 parameters to be sensitive. Another distinct difference is that PEST
Figure 5.8.  (a) Second order parameter interactions based on the RMSE measure computed using Sobol’s method. (b) Second order parameter interactions based on the TRMSE measure computed using Sobol’s method. Circles represent interactions that contribute at least 1% of the overall model output variance. The color legends and shading represent the Sobol indices’ magnitudes and ranges.
Figure 5.9.  (a) Comparative summary of sensitivity classifications based on the high-flow RMSE model performance objective.  (b) Comparative summary of sensitivity classifications based on the high-flow TRMSE model performance objective.

found the storage parameters (UZTWM, UZFWM, LZTWM, LZFSM, LZFPM) to be insensitive while other methods identified these parameters to be sensitive or highly sensitive. It is likely that the PEST results are reflective of local optima in the model’s response surface, which would be expected to be highly multimodal [97, 196].

Figure 5.9 shows that RSA generally defined the smallest subset of SAC-SMA/SNOW-17 parameters as being sensitive or highly sensitive. The RSA version used in this chapter is unique among the four tested sensitivity methods in the sense that our classifications required qualitative assessments of a visual representation of results. As noted above RSA yields very similar rankings for highly sensitive results, but the qualitative interpretation of sensitivity becomes more challenging for parameters that show modest sensitivity.

Although Fig. 5.9 provides a comparative synopsis of the different results
attained by the four sensitivity methods, it does not allow for any quantitative analysis of their relative effectiveness as screening tools. Building on Andres [181], we have tested the effectiveness of each of the sensitivity methods used in this chapter. We have used the sensitivity classifications given in Fig. 5.9 in combination with an independent LHS-based random draw of 1000 parameter groups for the 18 parameters analyzed in this study. Recall from Section 5.5.4 that the independent sample and the sensitivity classifications in Fig. 5.9 were used to develop three parameter sets. Set 1 consists of the full randomly generated independent sample set. In Set 2, the parameters classified as highly sensitive or sensitive are set to a priori fixed values while the remaining insensitive parameters are allowed to vary randomly. Lastly, in Set 3 the parameters classified as being highly sensitive or sensitive vary randomly and the insensitive parameters are set to a priori values.

Figure 5.10 illustrates that by plotting Set 2 versus Set 1 as well as Set 3 versus Set 1 we can test the effectiveness of the sensitivity analysis methods. As shown for the Sobol’s method and ANOVA results in Fig. 5.10a varying parameters that are correctly classified as “insensitive” in Set 2 should theoretically yield a zero correlation with the full random sample of Set 1 (i.e., plot as a horizontal line). If some parameters are incorrectly classified as insensitive then the scatter plots show deviations from a horizontal line and increased correlation coefficients as is the case for the PEST and RSA results in Fig. 5.10a. Conversely, if the correct subset of sensitive parameters is sampled randomly (i.e., Set 3) than they should be sufficient to capture model output from the random samples of the full parameter set in Set 1 yielding a linear trend with an ideal correlation coefficient of 1. Fig. 5.10b shows that the Sobol’s method yields the highest correlation between Set 3 and Set 1 followed closely by ANOVA. PEST yields the worst correlations for the hourly SPKP1 watershed’s results shown in Fig. 5.10. More generally, the plots in Fig. 5.10 show that this analysis can be quantified using correlation coefficients.

Table 5.13 provides a summary of correlation coefficients for all of the test cases analyzed in this study. PEST shows the worst performance overall and in a few cases PEST performed comparably to RSA for the RMSE-based evaluations. The correlation coefficients in Table 5.13 show that Sobol’s method and ANOVA perform very similarly in terms of their effectiveness. Sobol’s method has a slight advantage in that its sensitivity classifications are more certain or robust relative
Figure 5.10. Example illustration of an independent test of the sensitivity classifications found for the SPKP1 watershed’s model at the 1-hour timescale. The scatter plots show the RMSE of streamflow predictions. Set 1 consists of 1000 randomly drawn Latin hypercube samples. Set 2 is composed of constant reference values for sensitive parameters and random samples of the remaining insensitive parameters. In Set 3 sensitive parameters are allowed to vary randomly and insensitive parameters are set to constant reference values. Term $r$ represents correlation coefficient.

Table 5.13. Summary of correlation coefficients from the independent testing of each sensitivity method’s effectiveness.

<table>
<thead>
<tr>
<th>Sets</th>
<th>Tools</th>
<th>RMSE</th>
<th>SATP1</th>
<th>TRMSER</th>
<th>SATP1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1h</td>
<td>6h</td>
<td>24h</td>
<td>1h</td>
</tr>
<tr>
<td>Set2</td>
<td>PEST</td>
<td>0.748</td>
<td>0.749</td>
<td>0.782</td>
<td>0.282</td>
</tr>
<tr>
<td></td>
<td>RSA</td>
<td>0.184</td>
<td>0.242</td>
<td>0.264</td>
<td>0.029</td>
</tr>
<tr>
<td></td>
<td>ANOVA</td>
<td>0.059</td>
<td>0.046</td>
<td>0.050</td>
<td>-0.059</td>
</tr>
<tr>
<td></td>
<td>SOBOL</td>
<td>0.074</td>
<td>0.023</td>
<td>0.023</td>
<td>-0.013</td>
</tr>
<tr>
<td>Set3</td>
<td>PEST</td>
<td>0.191</td>
<td>0.316</td>
<td>0.272</td>
<td>0.426</td>
</tr>
<tr>
<td></td>
<td>RSA</td>
<td>0.822</td>
<td>0.734</td>
<td>0.831</td>
<td>0.699</td>
</tr>
<tr>
<td></td>
<td>ANOVA</td>
<td>0.979</td>
<td>0.981</td>
<td>0.984</td>
<td>1.000</td>
</tr>
<tr>
<td></td>
<td>SOBOL</td>
<td>0.991</td>
<td>0.985</td>
<td>0.950</td>
<td>1.000</td>
</tr>
</tbody>
</table>

to ANOVA’s sensitivity rankings. Sobol’s method and ANOVA have consistently superior correlations compared to both PEST and RSA, which should be expected given the importance of parameter interactions in the problems analyzed.

5.7 Discussion

The results of this chapter show that model parameter sensitivities are heavily impacted by the choice of analysis method as well as the selected model time interval. Differences between the two adjacent watersheds also suggest strong influences of local physical characteristics on the sensitivity methods’ results. As the only local sensitivity approach analyzed, it is not surprising that PEST yielded results that were often significantly different from the global sensitivity methods. As noted in Section 5.6.5, the most contradictory PEST result in this chapter was...
its classification of the SNOW-17 parameters and the SAC-SMA storage parameters as being sensitive. This apparent misclassification of the snow parameters reflects the biggest and to a degree most expected limitation for PEST. Readers should be aware that the linearization of the relationship between a model’s output and its parameters will adversely impact PEST applications for hydrologic models with thresholds because of their impacts on the derivatives in the Taylor’s series expansion. As a local sensitivity approach, PEST is more prone to misclassify sensitivities for highly multimodal response surfaces since the method’s derivatives are computed at a single point determined to be locally optimal by the Gauss-Marquardt-Levenberg algorithm [90, 197]. The model response surface is defined as the mapping of all sampled parameter inputs for a model into the set of RMSE (or TRMSE) values that define the accuracy of its performance. Hydrologic models have been shown to yield complex surfaces with a large number of local optima [53, 97, 196]. As the search space dimension and complexity increases, PEST’s composite Jacobean-based measure has an increasing likelihood of identifying sensitivities that represent local anomalies in the model’s response. PEST’s strengths lie in its computational efficiency, ease-of-implementation, and ease-of-use [for an application of inversion of a groundwater flow-and-transport model see [33]].

Overall the results in Table 5.13 indicate that RSA is more effective than PEST, but less effective than ANOVA and Sobol’s method. While RSA often identifies the same highly sensitive parameters as ANOVA and Sobol’s method, less sensitive parameters were often neglected. This reflects a methodological limitation of the RSA version we used where the qualitative interpretation of sensitivity becomes more challenging for parameters that show modest impacts on model performance. The qualitative nature of the RSA version we used prevents quantitative rankings of parameter sensitivities. Moreover, the qualitative interpretations ignore parameter interactions which further biases RSA to yield smaller “sensitive” parameter sets. The dimensionality of the parameter space being sampled in this study also seems to have a large impact on the RSA sensitivity classifications. As has been shown in numerous prior studies [53, 97, 125, 196], it is extremely challenging to identify regions of the SAC-SMA model’s response surface that are near optimal or optimal. Although the Latin hypercube sampling used in RSA showed statistical convergence at 10,000 samples, the plots used to classify parameters as being
sensitive (e.g., see Figs. 5.5 and 5.6) are guaranteed to under represent the true dispersion within their likelihood distributions due to the high-dimension of the sample space (i.e., 18 parameters). The strengths of RSA that have motivated its popularity in the hydrologic literature are its ease-of-implementation, its highly visual results, and its complementary support of uncertainty analysis.

ANOVA and Sobol’s method have a clear methodological distinction relative to PEST and RSA in that they readily account for 2nd order parameter interactions. Clearly, parameter interactions contribute to ANOVA and Sobol’s method classifying more parameters as being sensitive relative to RSA.

The F-values attained for ANOVA varied significantly in their computed confidence intervals, making the method’s ranking of moderately sensitive parameters highly uncertain. We found that Sobol’s parameter sensitivity rankings still showed variability, but that the overall rank order did not change as significantly as the ANOVA rankings. As noted by Mokhtari and Frey [93] and verified in this chapter, ANOVA is fairly robust for highly nonlinear models with thresholds, despite its normality assumptions. ANOVA’s use of the F-distribution in assigning sensitivities has positive and negative impacts on analysis. The F-values attained in ANOVA are easily ranked and sensitivity thresholds can be easily defined by users based on the likelihood of misclassification. Unfortunately, the F-values vary significantly with bootstrapping and sample size, impacting their rank ordering and meaningfulness as sensitivity metrics.

Alternatively, Sobol’s indices have very direct and meaningful interpretations in terms of the overall contribution to model output variance. Relative to the other methods tested, Sobol’s indices provided the most detailed description of how individual parameters and their interactions impact model performance. Although Sobol’s method is the most complex and computationally expensive [requiring $8,192 \times (2 \times 18 + 2) = 311,296$ model runs] sensitivity analysis technique tested, its robustness, ease-of-interpretation, and detailed results distinguish it among the methods tested in this study. Readers should note that the relatively small computational burden posed by the SAC-SMA/SNOW 17 model allowed us to be very conservative in our choice of sample size for Sobol’s method. Generally, the relative ranking of parameter sensitivities was stable for a much smaller sample size. Our use of 8,192 samples may be overly rigorous and reflects our minimization of
minor numerical errors that did not appreciably change sensitivity classifications.

5.8 Conclusions

This chapter tested four sensitivity analysis methods: (1) local analysis using parameter estimation software (PEST), (2) regional sensitivity analysis (RSA), (3) analysis of variance (ANOVA), and (4) Sobol’s method. These four sensitivity methods were applied to the lumped Sacramento soil moisture accounting model (SAC-SMA) coupled with SNOW-17. Results from this chapter characterize model sensitivities for the two medium sized watersheds within the Juniata River Basin in Pennsylvania, USA. Comparative results for the 4 sensitivity methods are presented for a 3-year time series with 1 hour, 6 hour, and 24 hour time intervals.

In this chapter, the SAC-SMA/SNOW-17 model parameters were classified into three categories: (1) highly sensitive, (2) sensitive, and (3) insensitive. The sensitivity results presented in Section 5.6 show that the fully lumped SAC-SMA/SNOW-17 model’s responses are “uniquely” determined by the performance objective specified, prediction timescale, and specific watershed being modeled. Generally, the global methods (RSA, ANOVA, and Sobol’s method) agreed on their classifications of the most sensitive parameters for each case. The local method, PEST, generated results that were often significantly different from the global sensitivity analysis methods.

PEST and RSA both neglect parameter interactions and as a consequence yield a far less nuanced description of the models they evaluate. In a broader context, sensitivity analysis shapes the manner in which hydrologists view the processes and watershed properties impacting their model results. The basic assumptions, used in PEST and RSA, such as neglecting parameter interactions, may manifest themselves in the subsequent myriad of potential uses of the hydrologic model (e.g., flood forecasting, observation network design, reservoir management, etc.) by providing an overly simplified view of the controls on a hydrologic system.

Overall ANOVA and Sobol’s method were shown to be superior to RSA and PEST. Relative to one another, ANOVA has reduced computational requirements and Sobol’s method yielded more robust sensitivity rankings. The results from Sobol’s method clearly show that second order parameter interactions explained
between 3 to 40% of the SAC-SMA/SNOW-17 model’s variance. The implication of this result is that a larger number of parameters and processes within the model are impacting its performance. This chapter shows that as prediction problems in hydrology grow in complexity, our analysis techniques need to evolve to better represent and understand how models behave.
Chapter 6

Study 4: Advancing the identification and evaluation of distributed rainfall-Runoff models using Sobol’s global sensitivity analysis

This chapter is drawn from Y. Tang et al.’s paper that is in review in Water Resources Research [160]. It provides a step-wise analysis of a conceptual grid-based distributed rainfall-runoff model, the United States National Weather Service (US NWS)’s Hydrology Laboratory Research Distributed Hydrologic Model (HL-RDHM). It evaluates model parameter sensitivities for annual, monthly, and event time periods with the intent of elucidating the key parameters impacting the distributed model’s forecasts. This chapter demonstrates a methodology that balances the computational constraints posed by global sensitivity analysis with the need to fully characterize the HL-RDHM’s sensitivities.
6.1 Introduction

Over the past decade the increasing availability of spatially distributed hydrometeorological data (e.g., precipitation, air temperature and soil properties) coupled with advances in computational resources has resulted in increasing interest in the development of spatially distributed hydrological models [e.g. 3–7]. Developers of distributed models seek to better simulate watershed behavior by taking advantage of spatially distributed forcing as well as distributed watershed parameters for a broader array of processes such as surface flow, groundwater flow, sediment transport, solute transport, etc. The increasing complexity of distributed models poses several challenges in terms of (1) their severe computational demands relative to lumped watershed models [8, 9], (2) their potential for over-parameterisation [10], and (3) their high dimensional, nonlinear parametric spaces and structural uncertainties [11].

For many current distributed hydrologic models, the number of model parameters can range from hundreds to several thousand per watershed model depending on the grid resolution and number of state predictions. These high dimensional parametric spaces make it extremely difficult to assess the impacts of parameters or combinations of the parameters on watershed model behavior. In an operational context, it becomes more challenging to characterize spatially distributed parameters directly using field measurements, which then yields very high dimensional model calibration problems [19, 20, 32, 34, 90, 97]. Early studies [14, 16] have highlighted that in the context of optimization, the hydrologic model calibration problem is ill-posed, often highly nonlinear, and multimodal (i.e., numerous local optima exist), especially for high-dimensional parameter spaces.

Model parameter sensitivity analysis has long been recognized as a helpful parameter screening tool that can be used to identify the key parameters controlling model performance [9, 11, 24, 25, 29, 30, 34, 80, 84, 87, 165, 193]. This is particularly important with the current shift towards distributed hydrologic models. Sieber and Uhlenbrook [30] have highlighted that sensitivity analysis can not only clarify the most important parameters but also help understand and improve the model structure potentially. However, to date, the computational demands and spatial complexity of distributed hydrologic models have limited our ability to
understand their parametric interactions and sensitivities. The limited body of recent literature applying sensitivity analysis to spatially distributed hydrologic models highlights the importance and significant challenges posed by this problem \cite{9, 11, 30, 32–34}.

Tang et al. \cite{160} comprehensively compared state-of-the-art sensitivity analysis tools including Sobol’s method, a Jacobian-based local method, regional sensitivity analysis, and Analysis of Variance (ANOVA). Sobol’s method \cite{94} was found to be the most effective approach in globally characterizing single parameter and multi-parameter interactive sensitivities. Building on this prior study, the present work utilizes Sobol’s global sensitivity analysis method to characterize the spatial and temporal variations of single parameter and multi-parameter interactions for the US NWS’s distributed hydrological watershed model termed HL-RDHM which is originally called Hydrology Laboratory Research Modeling System (HL-RMS) \cite{163, 188–190}. This chapter seeks to carefully characterize the HL-RDHM’s sensitivities while balancing the computational demands associated with Sobol’s method. These issues are important to both long-term and short-term forecasts. In the context of long-term forecasts, this chapter characterizes HL-RDHM’s sensitivities over annual time periods using 24-hour model time steps and monthly time periods using 1-hour model time steps. In these cases, computational demands required our use of distributed forcing and lumped model parameterizations. For event-level time periods, a detailed spatial analysis of the HL-RDHM’s sensitivities is presented for two events using hourly model time steps. The events were selected to explore how the spatial heterogeneity of forcing impacts the HL-RDHM’s spatial sensitivities.

In the remainder of this chapter, Section 6.2 provides an overview of the HL-RDHM distributed hydrologic model. Section 6.3 provides a detailed description of Sobol’s method as well as the statistical sampling scheme used in this study. Sections 6.4 and 6.5 present the case studies and the computational experiment used to characterize the HL-RDHM, respectively. Section 6.6 presents the sensitivity analysis results for annual, monthly, and event time-scales. Sections 6.7 discuss the implications of the HL-RDHM’s temporal and spatial sensitivity trends in terms of their value for simplifying model calibration and enhancing operational forecasting.
Figure 6.1. HL-RDHM model grid for the Saxton (SXTP1) and Spruce Creek (SPKP1) watersheds.

6.2 Overview of the Hydrology Laboratory Research Distributed Hydrologic Model (HL-RDHM)

6.2.1 Model structure

The HL-RDHM was developed by the US NWS [163, 188–190]. It is a flexible modeling framework for building lumped, semi-distributed, and fully distributed hydrological models. The structure of the modeling system is based on the Hydrologic Rainfall Analysis Project (HRAP) rectangular grid. The HRAP grid is defined at the $4\text{km} \times 4\text{km}$ resolution that corresponds directly to the US NWS’ Next Generation Weather Radar (NEXRAD) precipitation products. Figure 6.1 presents the HRAP grid cells for the Saxton (SXTP1) and Spruce Creek (SPKP1) headwaters for the Juniata river in Pennsylvania. SXTP1 and SPKP1 are the case study watersheds used to evaluate the HL-RDHM in this chapter. The arrows in the grid indicate the direction of surface flows. More information about HRAP gridded NEXRAD data can be found in Reed and Maidment [198].

For each HL-RDHM grid cell, a snow model, a rainfall-runoff model, and hills-
lope and channel routing models are used to simulate the rainfall-runoff processes. Fast model responses such as overland flow and direct runoff on impervious area are routed from the hillslope and drained into a conceptual channel which has a length of the cell diagonal distance. The inter-cell channel routing is conducted by using a connectivity file which reflects the surface flow directions as shown in Fig. 6.1. A modified version of the algorithm developed by Wang et al. [199] is used to generate the connectivity file [163]. Essentially, the fine-resolution Digital Elevation Model (DEM) cells were aggregated into the coarser HRAP grid cells. In the algorithm, the flow path defined from the DEM determines a grid cell’s flow direction. The algorithm constrains the grid flow directions to closely match the flow pattern that would be predicted using the high resolution DEM. The slow model responses such as interflow and baseflow go straight into the conceptual channel without going through hillslope routing.

In the version of the HL-RDHM used in this chapter, SNOW-17 [187] is used to simulate the energy balance of a snowpack using a temperature index method. Two models, the Sacramento Soil Moisture Accounting (SAC-SMA) model developed by Burnash [113] and the Continuous Antecedent Precipitation Index (CONT-API) developed by Anderson [200], are available for rainfall-runoff modeling. In this study, the SAC-SMA model is adopted because it is widely used by the river forecasting centers of the US NWS [188–190]. Hillslope and channel routing processes are modeled within HL-RDHM using a kinematic wave approximation to the St. Venant equations. Section 6.2.2 provides a more detailed description of the HL-RDHM’s component models.

6.2.2 The components of HL-RDHM

6.2.2.1 SNOW-17

SNOW-17 is a snow accumulation and ablation model initially developed by Anderson [187]. It requires precipitation and air temperature as its inputs and simulates the energy balance of the snow pack based on a temperature index method. The main processes modeled by SNOW-17 are snow melt, snow cover accumulation, surface energy exchange during non-melt periods, snow cover heat storage, areal extent of snow cover, retention and transmission of liquid water, and heat exchange.
Table 6.1. Summary of SNOW-17 and SAC-SMA parameters

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameters</th>
<th>Unit</th>
<th>Description</th>
<th>Allowable Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNOW-17</td>
<td>SCF</td>
<td></td>
<td>Gage catch deficiency adjustment factor</td>
<td>1.0-1.3</td>
</tr>
<tr>
<td></td>
<td>MFMAX</td>
<td>mm/°C/6h</td>
<td>Maximum melt factor during non-rain periods</td>
<td>0.5-1.2</td>
</tr>
<tr>
<td></td>
<td>MFMIN</td>
<td>mm/°C/6h</td>
<td>Minimum melt factor during non-rain periods</td>
<td>0.1-0.6</td>
</tr>
<tr>
<td></td>
<td>UADJ</td>
<td>mm/mb/6h</td>
<td>Average wind function during rain-on-snow periods</td>
<td>0.02-0.2</td>
</tr>
<tr>
<td></td>
<td>SI</td>
<td>mm</td>
<td>Mean water-equivalent above which 100% cover exists</td>
<td>10-120</td>
</tr>
<tr>
<td>SAC-SMA</td>
<td>UZTWM</td>
<td>mm</td>
<td>Upper zone tension water maximum storage</td>
<td>25.0-125.0</td>
</tr>
<tr>
<td></td>
<td>UZFWM</td>
<td>mm</td>
<td>Upper zone free water maximum storage</td>
<td>10.0-75.0</td>
</tr>
<tr>
<td></td>
<td>UZK</td>
<td>day⁻¹</td>
<td>Upper zone free water lateral depletion rate</td>
<td>0.2-0.5</td>
</tr>
<tr>
<td></td>
<td>PCTIM</td>
<td></td>
<td>Impervious fraction of the watershed area</td>
<td>0.0-0.01</td>
</tr>
<tr>
<td></td>
<td>ADIMP</td>
<td></td>
<td>Additional impervious area</td>
<td>0.0-0.02</td>
</tr>
<tr>
<td></td>
<td>ZPERC</td>
<td></td>
<td>Maximum percolation rate</td>
<td>20.0-300.0</td>
</tr>
<tr>
<td></td>
<td>REXP</td>
<td></td>
<td>Exponent of the percolation equation</td>
<td>1.4-3.5</td>
</tr>
<tr>
<td></td>
<td>LZTWM</td>
<td>mm</td>
<td>Lower zone tension water maximum storage</td>
<td>75.0-300.0</td>
</tr>
<tr>
<td></td>
<td>LZFSM</td>
<td>mm</td>
<td>Lower zone free water supplemental maximum storage</td>
<td>15.0-300.0</td>
</tr>
<tr>
<td></td>
<td>LZFP</td>
<td>mm</td>
<td>Lower zone free water primary maximum storage</td>
<td>40.0-600.0</td>
</tr>
<tr>
<td></td>
<td>LZSK</td>
<td>day⁻¹</td>
<td>Lower zone supplemental free water depletion rate</td>
<td>0.03-0.2</td>
</tr>
<tr>
<td></td>
<td>LZPK</td>
<td>day⁻¹</td>
<td>Lower zone primary free water depletion rate</td>
<td>0.001-0.015</td>
</tr>
<tr>
<td></td>
<td>PFREE</td>
<td></td>
<td>Fraction of water percolating from upper zone directly to lower zone free water storage</td>
<td>0.0-0.5</td>
</tr>
</tbody>
</table>

at the snow-soil interface.

Snow melt is separated for rain-on-snow periods and non-rain periods. The energy and mass balance equations, in which the average wind function (UADJ) is the only parameter, are used to calculate the snow melt during rain-on-snow periods. Empirical equations, in which the maximum melt factor (MFMAX) and the minimum melt factor (MFMIN) are control parameters, are utilized to compute the snow melt during non-rain periods. The threshold temperature PXTEMP determines the precipitation form when calculating the accumulation of snow cover. The snowfall correction factor (SCF) is used to adjust the biased estimations of precipitation during snowfall. The areal extent of snow cover is determined by a depletion curve which describes the relationship between the areal extent of snow and its areal water equivalent according to the historical maximum water equivalent and the threshold of water equivalent for 100% snow coverage. More details about the main processes and key parameters of SNOW-17 are available in Anderson [187]. Based on the prior analysis of Tang et al. [160], we have focused our sensitivity analysis on the five SNOW-17’s parameters listed in Table 6.1. The feasible ranges of these five parameters were defined based on the recommendations of Anderson [191].
6.2.2.2 Sacramento Soil Moisture Accounting (SAC-SMA) model

The SAC-SMA model \cite{113} is a lumped conceptual hydrological model that has been commonly employed by the US NWS \cite{163}. In the model, the soil is characterized by a thinner upper zone and a thicker lower zone. Both the upper zone and lower zone are partitioned into free water storage and tension water storage. Surface runoff is generated based on the saturation-excess as well as the infiltration-excess mechanisms \cite{113}. The maximum upper zone tension water storage (UZTWM) and the maximum upper zone free water storage (UZFWM) must be filled up before surface runoff can take place. The excess water overflows into the upper zone free water storage after UZTWM is filled. The upper zone free water percolates to the lower zone and can also spill out horizontally as interflow when the precipitation rate exceeds the percolation rate. Surface runoff occurs when the precipitation rate exceeds the percolation rate and UZFWM is filled. The percolation is determined by the upper zone free water content and the percolation demand which is a function of the deficiency of lower zone water storages. Before the LZTWM is filled, a fraction of percolated water goes into lower zone tension water storage while the remainder enters the lower zone free water storages. The fraction of percolated water entering into the lower zone free water is represented by parameter PFREE. All of the percolated water goes into lower zone free water after LZTWM is filled. Unlike the upper zone, the free water in the lower zone consists of primary and supplemental free water storages. They generate base-flow independently at different drainage rates based on Darcy’s law. The capacities for the lower zone free water, the primary free water, and the supplemental free water are represented by the parameters LZTWM, LZFPM, and LZFSM, respectively. Direct runoff, surface runoff, interflow, and baseflow are the four major forms of runoff in the SAC-SMA model outputs. The permanent and temporal impervious area contribute to the direct runoff. These two areas are defined using the two parameters termed percent of impervious area (PCTIM) and additional impervious area (ADIMP), respectively. The permanent impervious area is fixed during model runs. The additional impervious area (ADIMP) varies dynamically during the simulation as the upper zone tension water fills causing the temporal seepage outflow areas to become impervious temporarily.

Thirteen SAC-SMA model parameters were selected for analysis in this study.
The feasible ranges of these parameters are listed in Table 6.1 and are based on the recommendations of Anderson [191].

### 6.2.2.3 Hillslope and channel routing model

In HL-RDHM, both hillslope and channel routing use a kinematic wave approximation to the St. Venant equations, i.e., the continuity and momentum equations. The general form of the continuity equation is defined as

$$\frac{\partial Q}{\partial x} + \frac{\partial A}{\partial t} = q$$  \hspace{1cm} (6.1)

where $A$ and $Q$ are cross-section area and discharge, respectively. The lateral inflow, $q$, varies for hillslope routing applications versus channel routing applications. In hillslope routing, $q$ is the fast runoff from the rainfall-runoff model. In contrast, for channel routing, $q$ is the summation of the routed overland flow and the slow runoff from the rainfall-runoff model. The hillslope and channel routing components use slightly different forms of the simplified momentum equation (storage-release equation for the kinematic wave approximation). At the hillslope edges, the boundary condition is assumed to be zero flow. For channel routing, the upper boundary condition is the total discharge of all the grid cells which drain to a cell [163].

Hillslope slope, hillslope roughness coefficient, and drainage density are the principal parameters for hillslope routing. The hillslope slope can be estimated from DEM data. Koren et al. [163] state that a spatially constant value of the hillslope roughness coefficient is sufficient. The two major parameters of channel routing, specific discharge and the exponent parameter can be estimated using either the channel shape method or the rating curve method following the recommendations of Koren et al. [163].
6.3 Sobol’s sensitivity analysis

6.3.1 Sobol’s method

Sobol’s method is a variance-based sensitivity analysis approach that represents a model in the following functional form:

\[ y = f(x, \Theta) \]  

(6.2)

where \( y \) is the model outputs, \( x \) is the input state variables, and \( \Theta \) is the parameter set. Sobol’s method evaluates parametric sensitivity by evaluating the variance of \( y \) due to changes of parameter vector \( \Theta \). As described by Sobol’ [94], the total variance of the model output is decomposed into component variances that result from individual parameters and parameter interactions. Typically, the direct model output \( y \) is replaced by a model performance measure such as the root mean square error (RMSE) of the streamflow model predictions. Single parameters or parameter interactions are then evaluated according to their percentage contribution to the total variance of model responses. The Sobol’s variance decomposition shown in Eq. 6.3 assumes that the parameters are independent:

\[ D(y) = \sum_i D_i + \sum_{i<j} D_{ij} + \sum_{i<j<k} D_{ijk} + D_{12...m} \]  

(6.3)

where \( D_i \) is the measure of the sensitivity to model output \( y \) due to the \( i^{th} \) component of the input parameter vector denoted as \( \Theta \), \( D_{ij} \) is the portion of output variance due to the interaction of parameters \( \theta_i \) and \( \theta_j \). The variable \( m \) stands for the total number of parameters. Sobol’s sensitivity indices are computed using the following equations derived from Eq. 6.3

\[
\text{first order } S_i = \frac{D_i}{D} \tag{6.4}
\]

\[
\text{total order } S_{Ti} = 1 - \frac{D_{\sim i}}{D} \tag{6.5}
\]

where \( S_i \) indicates the sensitivity that results from the main effect of parameter \( \theta_i \). The average variance, \( D_{\sim i} \), results from all of the parameters except for \( \theta_i \). The total order sensitivity, \( S_{Ti} \), defines the independent and interactive effects up to
the \( m^{th} \) order of parameter \( \theta_i \). A parameter mainly influences the model output by parameter interactions if it has a small first order index and a large total order sensitivity index.

When implementing Sobol’s method, the parameter ranges are scaled to be between 0 and 1. Equation 6.2 is partitioned as follows:

\[
f(\theta_1, \ldots, \theta_m) = f_0 + \sum_{i=1}^{m} f_i(\theta_i) + \sum_{1 \leq i < j \leq m} f_{ij}(\theta_i, \theta_j) + \cdots + f_{1,2,\ldots,m}(\theta_1, \ldots, \theta_m) \tag{6.6}
\]

and the normalized total variance \( D \) can be evaluated by Eq. 6.7:

\[
D = \int_{0}^{1} \cdots \int_{0}^{1} f(\Theta)d\Theta - f_0^2 \tag{6.7}
\]

and the variances \( D_{i_1,\ldots,i_s} \) are,

\[
D_{i_1,\ldots,i_s} = \int_{0}^{1} \cdots \int_{0}^{1} f^2_{i_1,\ldots,i_s}(\theta_{i_1}, \ldots, \theta_{i_s})d\theta_{i_1}d\theta_{i_s} \tag{6.8}
\]

\[1 \leq i_1 < \cdots < i_s \leq m, \quad s = 1, \ldots, m \tag{6.9}\]

When the model is highly nonlinear and complex, Monte Carlo numerical integration is the most suitable method for evaluating the integrals represented in equations (7)-(9) above. The Monte Carlo approximations for \( D, D_i, \) and \( D_{i,\sim,s} \) are given in the following equations as presented in prior studies by Sobol’ [94, 95] and Hall et al. [28]:

\[
\hat{f}_0 = \frac{1}{n} \sum_{s=1}^{n} f(\Theta_s) \tag{6.10}
\]

\[
\hat{D} = \frac{1}{n} \sum_{s=1}^{n} f^2(\Theta_s) - \hat{f}_0^2 \tag{6.11}
\]

\[
\hat{D}_i = \frac{1}{n} \sum_{s=1}^{n} f(\Theta_s^{(a)})f(\Theta_s^{(b)}; \Theta_{i_s}^{(a)}) - \hat{f}_0^2 \tag{6.12}
\]

\[
\hat{D}_{i,\sim,s} = \frac{1}{n} \sum_{s=1}^{n} f(\Theta_s^{(a)})f(\Theta_s^{(a)}; \Theta_s^{(b)}) - \hat{f}_0^2 \tag{6.13}
\]

In equations (10)-(13), the variable \( n \) defines the Monte Carlo sample size, \( \Theta_s \).
represents the sampled individual in the scaled unit hypercube, and \((a)\) and \((b)\) are two different samples. The parameters with values drawn from sample \((a)\) are denoted by \(\Theta_{is}^{(a)}\). The variables \(\Theta_{is}^{(a)}\) and \(\Theta_{is}^{(b)}\) designate that parameter \(\theta_i\) draws values from sample \((a)\) and \((b)\), respectively. The symbols \(\Theta_{(\sim_i)s}^{(a)}\) and \(\Theta_{(\sim_i)s}^{(b)}\) define cases when parameter \(\theta_i\) is not sampled and the remaining parameters do get their sample values from samples \((a)\) and \((b)\), respectively.

Although Sobol’s method can identify important parameter interactions, the method becomes computationally expensive when high order interactions (or indices) must be calculated for models with large parameter sets (e.g. distributed hydrologic models). The original Sobol’s method [94] required \(n \times (2m + 1)\) model runs to calculate all of the first order and the total order sensitivity indices. Recall that \(n\) is the number of Monte Carlo samples and \(m\) is the number of model parameters. An enhancement of the method made by Saltelli [183] provides the first, second, \((m-2)^{th}\), and total order sensitivity indices using \(n \times (2m + 2)\) model runs. Saltelli [183] provides an alternate method to calculate the first, \((m-2)^{th}\), and total order sensitivity at a reduced cost of \(n \times (m + 2)\) model simulations. For this study, the latter method is desirable because it not only reduces the run time for analysis but it also provides sufficient information regarding main effects and parameter interactions.

### 6.3.2 Latin Hypercube Sampling (LHS)

In this study, Latin hypercube sampling (LHS) was used to sample the feasible parameter space because it was found to be effective in prior studies [e.g. 30, 160, 174]. LHS integrates the strength of random sampling and stratified sampling [76, 175] to make sure that all portions of the parameter space are considered. The method divides the \(m\) parameters’ ranges into \(n\) disjoint intervals with equal probability \(1/n\) from which one value is sampled randomly in each interval. To create one sample, a value of one parameter from a specific interval is picked up and combined randomly with a value from another parameter from an interval, then this pair of values are combined with a value of the third parameter and the process repeated till each parameter has a value in the combined parameter set. The values which have been picked to form an individual parameter draw do not
participate in generating other individuals (i.e., sampling without replacement). After all values of the variables have been chosen to create individuals, a sample of size \( n \) is created. The process can be repeated \( r \) times by so that a sample of total size \( r \times n \) is created. A Major benefit of the LHS method is its ability to divide parameter spaces into hypercubes to ensure a well spread parameter sample. More details about LHS are available in the following papers [76, 175, 176].

6.4 Case study

6.4.1 Juniata watershed description

Two headwater watersheds SPKP1 (drainage area 570 km\(^2\), mean elevation 485 m) and SXTP1 (drainage area 1960 km\(^2\), mean elevation 457 m) contributing to the Juniata River where used as test cases in this chapter. The watershed boundaries and the associated HL-RDHM model grids are shown in Fig. 6.1. The Juniata River Basin has a drainage area of 8800 km\(^2\) in south central Pennsylvania, USA, and is a major tributary to the Susquehanna River. As highlighted by Tang et al. [160], the SPKP1 and SXTP1 watersheds have different hydrologic conditions and basin characteristics which define their response behavior [also see [160, 188, 201]].

6.4.2 Data set

The input forcing data of the HL-RDHM consist of precipitation, monthly potential evapotranspiration (PE), and air temperature. The precipitation data are based on the NEXRAD multisensor precipitation estimator data developed on a HRAP grid with a 4km \( \times \) 4km resolution as shown in Fig. 6.1. The selected data set for precipitation and air temperature were attained from the US NWS mid-Atlantic river forecast center (MARFC) and are defined for an hourly time interval over the three year period spanned from January 1\(^{st}\), 2001 to December 31\(^{st}\), 2003. The observed streamflow in the same period was obtained from United States Geological Survey (USGS) gauge stations located at the outlets of the SPKP1 and SXTP1 watersheds. The time series of the 3 years’ precipitation and observed streamflow are presented in Fig. 6.2. The statistics of the precipitation and streamflow presented in Table 6.2 show that 2001 was the driest year and 2003 was the wettest
Figure 6.2. Hydrographs for the two studied watersheds from year 2001 to year 2003. (a) Hydrograph for SPKP1. (b) Hydrograph for SXTP1.

Table 6.2. Statistics of the precipitation and streamflow data from year 2001 to year 2003

<table>
<thead>
<tr>
<th></th>
<th>SPKP1</th>
<th>SXTP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precipitation</td>
<td>794.46</td>
<td>947.69</td>
</tr>
<tr>
<td>Runoff (mm)</td>
<td>377.45</td>
<td>496.93</td>
</tr>
<tr>
<td>Runoff/Precipitation</td>
<td>0.48</td>
<td>0.52</td>
</tr>
</tbody>
</table>

year among the three years. Both years 2001 and 2002 had a similar rainfall-runoff ratio. Table 6.3 illustrates the percentages of time when air temperature is below zero and precipitation is larger than zero based on the mean area precipitation and temperature. It reveals that frozen conditions generally occur from October to December and from January to April.

6.5 Computational experiment

6.5.1 Model setup

The current version of the HL-RDHM program is a mixed code developed with the FORTRAN, C, and C++ languages. We ran the model on a Linux computing
Table 6.3. Percentage of time when air temperature is below zero and precipitation is larger than zero. \( \text{Tair} \) denotes air temperature and \( P \) defines precipitation. The statistics is based on hourly mean area precipitation and air temperature.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>79.8</td>
<td>49.1</td>
<td>88.8</td>
<td>81.5</td>
<td>50.7</td>
<td>90.3</td>
<td>5.6</td>
<td>5</td>
<td>10.2</td>
<td>9.1</td>
<td>5.2</td>
<td>12.2</td>
</tr>
<tr>
<td>2</td>
<td>58</td>
<td>56.4</td>
<td>87.5</td>
<td>58.9</td>
<td>51</td>
<td>92</td>
<td>3.4</td>
<td>6.5</td>
<td>20.2</td>
<td>4.2</td>
<td>2.8</td>
<td>21</td>
</tr>
<tr>
<td>3</td>
<td>48.3</td>
<td>28.9</td>
<td>37.9</td>
<td>50.7</td>
<td>30.5</td>
<td>38.7</td>
<td>9.1</td>
<td>4.4</td>
<td>7.6</td>
<td>8.6</td>
<td>4.6</td>
<td>5.6</td>
</tr>
<tr>
<td>4</td>
<td>10</td>
<td>10.8</td>
<td>10.7</td>
<td>10</td>
<td>10</td>
<td>9.9</td>
<td>2.1</td>
<td>0.6</td>
<td>2.5</td>
<td>2.1</td>
<td>0.8</td>
<td>2.2</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>4.8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>10</td>
<td>4.8</td>
<td>3.4</td>
<td>2.4</td>
<td>7.3</td>
<td>5</td>
<td>9.7</td>
<td>0</td>
<td>1.3</td>
<td>0.3</td>
<td>0</td>
<td>1.2</td>
<td>0.3</td>
</tr>
<tr>
<td>11</td>
<td>11.7</td>
<td>23.3</td>
<td>17.5</td>
<td>15</td>
<td>27.5</td>
<td>22.6</td>
<td>0.6</td>
<td>4.6</td>
<td>1.8</td>
<td>0.6</td>
<td>6.1</td>
<td>3.3</td>
</tr>
<tr>
<td>12</td>
<td>44.5</td>
<td>74.1</td>
<td>69.5</td>
<td>46.1</td>
<td>73.4</td>
<td>75</td>
<td>2.6</td>
<td>12.2</td>
<td>13.3</td>
<td>2</td>
<td>12.6</td>
<td>13.7</td>
</tr>
</tbody>
</table>

cluster with 133 computer nodes composed of dual or quad AMD Opteron processors and 64 GB of RAM. Global sensitivity analysis is often considered to be intractable for complex models with long computation times because thousands of model simulations are needed \([\text{e.g.} \ [160]\]). To facilitate the analysis presented in this chapter, modifications have been made to HL-RDHM’s file input and output so that the forcing data and intermediate results are stored in dynamically allocated memory during the analysis. Additionally, we further enhanced the computational tractability of our analysis by parallelizing our implementation of Sobol’s method. The parallel program was written for distributed memory clusters using the Message Passing Interface (MPI) \([65]\). The cluster used in this research runs the GNU/Linux operating system and the Argonne National Laboratory’s Infinicon Message Passing Interface (MPI). The Infinicon MPI runs natively over a very high-bandwidth, ultra low-latency network interconnect. This application utilized up to 8 processors for single case studies.

6.5.2 Test cases and parameterization

Computation time poses a severe constraint for analyzing the HL-RDHM’s sensitivity. Consequently, we have carefully designed our analysis as a step-wise progression from long-term to short-term time periods to maintain the computational tractability of Sobol’s method. Three test cases were configured based on these considerations: (1) Annual sensitivity analysis using 3 years of observations and
Figure 6.3. Hydrographs for the two analyzed events in SPKP1. (a) Hydrograph for May 2002 event. (b) Hydrograph for September 2003 event.

precipitation data modeled using daily time steps for the SPKP1 and SXTP1 watersheds, (2) Monthly sensitivity analysis using 3 years observation and precipitation data modeled using hourly time steps for the SPKP1 and SXTP1 watersheds, and (3) Event sensitivity analysis using two events with hourly observed data and model time steps for the SPKP1 watershed. For the first 2 test cases, the parameters of the HL-RDHM were spatially lumped (i.e., every grid cell takes the same value for a specific parameter) in order to make Sobol’s method computationally tractable. For test case (3), the analysis was conducted using spatially distributed parameters (i.e., the parameter value varied from one model cell to another for a specific parameter). In test case (3), only 13 SAC-SMA parameters were analyzed because no snow occurred during these two events. For all of the test cases, the precipitation and the air temperature were spatially distributed. The hydrographs of the two selected events for test case (3) are presented in Fig. 6.3.

The period between January 1st and February 28th, 2001 were used as the warmup periods for the first 2 test cases to limit the impact of initial conditions. Similarly, a 2 month warmup period was used for both the May 2002 event and the September 2003 events as shown in Fig. 6.3. The a priori parameter settings used for the SNOW-17, SAC-SMA, and routing models were set based on the recommendations of the MARFC of the US NWS.

6.5.3 Sensitivity analysis implementation

Statistical sample size is a key parameter for Sobol’s method. In this study, the sample sizes were configured based on both literature recommendations and exper-
iments that tested the convergence and reproducibility of the sensitivity analysis results. In this study, LHS replaced the Sobol’s quasi-random sequence sampling used in our prior study because Sobol’s sequence sampling can sample a maximum of 100 parameters, which is insufficient given that test case (3) has 403 parameters (31 cells × 13 parameters). In our prior study, an extremely conservative sample size of 8,192 was used. However, in this study, we can not afford as many model runs as the prior study because the model execution time is significantly higher when switching from a lumped to a distributed model. A LHS sample size of 2000 was used in this study for all three test cases resulting in $2000 \times (18 + 2) = 40,000$ model runs per watershed for the first 2 test cases and $2000 \times (403 + 2) = 810,000$ per event model runs for the third test case. The results of Section 6.6.4 confirm that this sample size is sufficient to maintain the accuracy and repeatability of Sobol’s method. The most computationally intensive cases were the monthly analysis in the SXTP1 watershed (113 cells) and the single event (May 2002 or September 2003) analysis in SPKP1 (31 cells). Their estimated evaluation times are about 28 days and 16 days respectively on a single processor, which demonstrates the importance of using parallel computing to support this analysis.

The bootstrap method was used to provide confidence intervals for the parameter sensitivity rankings for the Sobol’s method. Essentially, the samples generated by LHS were resampled $N$ times when calculating the sensitivity indices for each parameter, resulting in a distribution of the indices. The moment method was used for attaining the bootstrap confidence intervals (BCIs). The moment method is based on large sample theory and requires a sufficiently large resampling dimension to yield symmetric 95% confidence intervals. In this study, the resample dimension $N$ was set to 2,000 based on prior literature discussions and our prior study. Readers interested in detailed descriptions of the bootstrapping method used in this chapter can check the referenced sources.
6.6 Results

Sections 6.6.1-6.6.4 present a step-wise analysis of the HL-RDHM’s sensitivities from annual to event level time-scales with the intent of elucidating the key parameters impacting the model’s forecasts. As discussed in the previous section, this chapter seeks to carefully characterize the HL-RDHM sensitivities while balancing the computational demands associated with Sobol’s sensitivity analysis. In the context of long-term forecasts, Sections 6.6.1 and 6.6.2 focus on HL-RDHM’s sensitivities for annual and monthly periods where computational demands required our use of distributed forcing and lumped model parameterizations for the SPKP1 and SXTP1 watersheds. Alternatively, Section 6.6.3 provides a detailed spatial analysis of the HL-RDHM’s sensitivities for the SPKP1 watershed for two events. These events were selected to explore how the spatial heterogeneity of forcing impacts the model’s spatial sensitivities. Section 6.6.4 continues our spatial analysis of event sensitivities and evaluates how well Sobol’s sensitivity method performs in identifying the principle input parameters controlling the HL-RDHM’s response. Our evaluation of Sobol’s method extends the SA repeatability test recommended by Anderson [181] to a spatially distributed modeling context.

6.6.1 Annual sensitivities based on distributed forcing and lumped parameters

Long-term distributed model runs pose significant computational demands and currently limit the tractability of detailed spatial analysis of parametric sensitivity. Recall from Section 6.3.1 that Sobol’s method requires \( n(m + 2) \) model runs to compute first and total order sensitivity indices. In this section, the computational demands associated with Sobol’s method were made tractable by lumping the HL-RDHM’s parameters (i.e., all cells had the same parameter values) while maintaining spatially distributed forcing and model structure. The first order and total order Sobol’s indices are reported in Tables 6.4 and 6.5 respectively, for the 18 parameters analyzed. Recall that the first order indices measure single parameter contributions to the HL-RDHM’s output variance, whereas the total order indices also include the influence of parameter interactions. Highly sensitive parameters are designated with dark grey shading, sensitive parameters have light grey shad-
<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter</th>
<th>SPKP1</th>
<th>SXTP1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNOW-17</td>
<td>SCF</td>
<td>0.00 [0.01]</td>
<td>0.00 [0.00]</td>
</tr>
<tr>
<td></td>
<td>MFMAX</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
</tr>
<tr>
<td></td>
<td>MFMIN</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
</tr>
<tr>
<td></td>
<td>UADJ</td>
<td>0.00* [0.00]</td>
<td>0.00* [0.00]</td>
</tr>
<tr>
<td></td>
<td>SI</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
</tr>
<tr>
<td></td>
<td>UZTWM</td>
<td>0.19 [0.03]</td>
<td>0.06 [0.02]</td>
</tr>
<tr>
<td></td>
<td>UZFWM</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.01]</td>
</tr>
<tr>
<td></td>
<td>UZK</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
</tr>
<tr>
<td></td>
<td>PCTIM</td>
<td>0.16 [0.02]</td>
<td>0.04 [0.01]</td>
</tr>
<tr>
<td></td>
<td>ADIMP</td>
<td>0.00 [0.01]</td>
<td>0.07 [0.02]</td>
</tr>
<tr>
<td></td>
<td>ZPERC</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.01]</td>
</tr>
<tr>
<td>SAC-SMA</td>
<td>REXP</td>
<td>0.00* [0.00]</td>
<td>0.00* [0.01]</td>
</tr>
<tr>
<td></td>
<td>LZTWM</td>
<td>0.06 [0.02]</td>
<td>0.49 [0.04]</td>
</tr>
<tr>
<td></td>
<td>LZFSM</td>
<td>0.03 [0.01]</td>
<td>0.02 [0.01]</td>
</tr>
<tr>
<td></td>
<td>LZFFPM</td>
<td>0.04 [0.02]</td>
<td>0.33 [0.03]</td>
</tr>
<tr>
<td></td>
<td>LZSK</td>
<td>0.01 [0.01]</td>
<td>0.01 [0.01]</td>
</tr>
<tr>
<td></td>
<td>LZPK</td>
<td>0.02 [0.01]</td>
<td>0.02 [0.01]</td>
</tr>
<tr>
<td></td>
<td>PFREE</td>
<td>0.34 [0.04]</td>
<td>0.02 [0.01]</td>
</tr>
</tbody>
</table>

* Negative mean value was set to zero

Table 6.4. Annual first order sensitivity indices from Sobol’s method computed using the RMSE measure and 24-hour model time steps. Dark gray shading designates highly sensitive parameters defined using a threshold value of 0.1. Light gray designates sensitive parameters defined using a threshold value of 0.01. White cells in the table designate insensitive parameters. The values in the brackets provide the 95% confidence interval for the indices’ values (i.e., the unbracketed value ± the bracketed value yields the confidence interval).
Table 6.5. Annual total order sensitivity indices from Sobol’s method computed using the RMSE measure and 24-hour model time steps. Dark gray shading designates highly sensitive parameters defined using a threshold value of 0.1. Light gray designates sensitive parameters defined using a threshold value of 0.01. White cells in the table designate insensitive parameters. The values in the brackets provide the 95% confidence interval for the indices’ values (i.e., the unbracketed value ± the bracketed value yields the confidence interval).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SOW-17</td>
<td>SCF</td>
<td>0.01 [0.01]</td>
<td>0.00 [0.00]</td>
<td>0.03 [0.01]</td>
<td>0.00 [0.01]</td>
<td>0.00 [0.00]</td>
<td>0.02 [0.01]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>MFMAX</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.01 [0.01]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.01 [0.01]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>MFMIN</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.03 [0.01]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.03 [0.01]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>UADJ</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.01 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.01 [0.00]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>SI</td>
<td>0.00* [0.00]</td>
<td>0.00* [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.00* [0.00]</td>
<td>0.00* [0.00]</td>
<td>0.00 [0.00]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>UZTWM</td>
<td>0.22 [0.03]</td>
<td>0.07 [0.02]</td>
<td>0.01 [0.01]</td>
<td>0.18 [0.03]</td>
<td>0.06 [0.01]</td>
<td>0.01 [0.01]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>UZFWM</td>
<td>0.00 [0.00]</td>
<td>0.01 [0.01]</td>
<td>0.02 [0.01]</td>
<td>0.00 [0.01]</td>
<td>0.00 [0.00]</td>
<td>0.02 [0.01]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>UZK</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.01 [0.01]</td>
<td>0.00 [0.00]</td>
<td>0.00 [0.00]</td>
<td>0.01 [0.01]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>PETITM</td>
<td>0.16 [0.02]</td>
<td>0.04 [0.01]</td>
<td>0.08 [0.02]</td>
<td>0.14 [0.02]</td>
<td>0.06 [0.01]</td>
<td>0.08 [0.02]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>ADIMP</td>
<td>0.02 [0.01]</td>
<td>0.07 [0.02]</td>
<td>0.21 [0.03]</td>
<td>0.03 [0.01]</td>
<td>0.06 [0.02]</td>
<td>0.26 [0.03]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>ZPERC</td>
<td>0.01 [0.01]</td>
<td>0.02 [0.01]</td>
<td>0.05 [0.02]</td>
<td>0.03 [0.02]</td>
<td>0.02 [0.01]</td>
<td>0.06 [0.02]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>REXP</td>
<td>0.00 [0.00]</td>
<td>0.01 [0.01]</td>
<td>0.05 [0.01]</td>
<td>0.00 [0.00]</td>
<td>0.01 [0.00]</td>
<td>0.05 [0.01]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>LZTWM</td>
<td>0.11 [0.02]</td>
<td>0.57 [0.04]</td>
<td>0.21 [0.02]</td>
<td>0.13 [0.03]</td>
<td>0.60 [0.04]</td>
<td>0.18 [0.02]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>LZFSM</td>
<td>0.04 [0.02]</td>
<td>0.04 [0.01]</td>
<td>0.15 [0.09]</td>
<td>0.07 [0.02]</td>
<td>0.03 [0.01]</td>
<td>0.20 [0.03]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>LZFPN</td>
<td>0.04 [0.02]</td>
<td>0.15 [0.03]</td>
<td>0.22 [0.03]</td>
<td>0.08 [0.02]</td>
<td>0.18 [0.02]</td>
<td>0.18 [0.03]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>LZSK</td>
<td>0.02 [0.01]</td>
<td>0.03 [0.01]</td>
<td>0.08 [0.02]</td>
<td>0.05 [0.02]</td>
<td>0.02 [0.01]</td>
<td>0.08 [0.02]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>LZPK</td>
<td>0.03 [0.01]</td>
<td>0.03 [0.01]</td>
<td>0.04 [0.02]</td>
<td>0.02 [0.01]</td>
<td>0.03 [0.01]</td>
<td>0.05 [0.02]</td>
</tr>
<tr>
<td>SOW-17</td>
<td>PFREE</td>
<td>0.40 [0.04]</td>
<td>0.03 [0.01]</td>
<td>0.01 [0.01]</td>
<td>0.36 [0.04]</td>
<td>0.12 [0.02]</td>
<td>0.01 [0.01]</td>
</tr>
</tbody>
</table>

* Negative mean value was set to zero

SAC-SMA/SNOW-17 sensitivity analysis of Tang et al. [160] for the same two watersheds and the same set of analyzed parameters (see Chapter 5). Tang et al. [160] concluded that each watershed had a unique set of sensitivities for the fully lumped SAC-SMA/SNOW-17 model. In contrast, the distributed forcing/lumped parameter results in Tables 6.4 and 6.5 show nearly identical sensitivity classifications for the two watersheds for each given year when spatially distributed forcing is considered.

The most significant changes in sensitivity for the watersheds occurred in the transition from dry conditions to wet conditions. Under the dry conditions of 2001, the HL-RDHM’s upper zone storages had to be filled before significant interflow and/or subsurface flow was possible. The first order and total order indices in Tables 6.4 and 6.5 show that the upper zone tension water storage (UZTWM), impervious cover (PCTIM), and the fraction of water percolating from the upper to the lower zone (PFREE) control 60 to 70 percent of the HL-RDHM’s response. As
would be expected the fraction of percolating water (PFREE) is highly interactive during the dry conditions of 2001 with approximately 6-percent of its influence on model output coming from interactions with other parameters.

In the 2002 transitional year, Tables 6.4 and 6.5 show a strong shift in parametric sensitivity from the upper zone storage to the lower zone storage. In the transitional year, the maximum storage in the lower zone tension water (LZTWM) and free water (LZFPM) influence about 60-percent of the HL-RDHM’s response. Comparing Tables 6.4 and 6.5 for 2002, shows that LZTWM is highly interactive and influences an additional 10-percent of the model’s variance with its interactions with other parameters. Conceptually, this makes sense given that 2002 is in general a wetting period for the subsurface in both watersheds.

The wet conditions in 2003 shift HL-RDHM’s sensitivities to being nearly equally distributed between all of the lower zone storage parameters (LZTWM, LZFSM, LZFPM) in both watersheds. Additionally, the wet conditions of 2003 also increased the incidence of surface saturation, which is reflected in the model’s sensitivity to additional impervious area (ADIMP). Overall in 2003, LZTWM, LZFSM, LZFPM, and ADIMP explain approximately 80-percent of the HL-RDHM’s output variance. The next section further elucidates HL-RDHM’s changes in sensitivity across the transition from dry to wet conditions using a per-month analysis.

6.6.2 Monthly sensitivities based on distributed forcing and lumped parameters

The results shown in Figure 6.4 provide a more detailed description of the temporal trends in the HL-RDHM’s sensitivities. These plots classify highly sensitive and sensitive parameters on a monthly basis using lumped parameters and distributed forcing at an hourly time step. The magnitude and ranges of Sobol’s indices are shown by the color legends. In the prior section, the transition from dry conditions in 2001 to wet conditions in 2003 resulted in a shift in the dominant parameters controlling the HL-RDHM response from the upper zone storage (UZTWM), impervious cover (PCTIM), and percolating water (PFREE) to lower zone storages (LZTWM, LZFSM, LZFPM). Figure 6.4 shows that these parameters’ are dominant in a year when they are classified as highly sensitive over a majority of
its component months. Although this result is expected, it is interesting to note the temporal transitions in sensitivity from the dry conditions in 2001 to the wet conditions in 2003.

For example, note that the upper zone storage (UZTWM) strongly influences the HL-RDHM response from March 2001 to February 2002. From January 2002 to March 2002, UZTWM’s total order indices [see Figure 6.4b] decrease from approximately 0.6 in both watersheds to approximately 0.2 (a three-fold decrease). A similar trend exists for water percolating from the model’s upper zone to its lower zone (i.e., PFREE). Note in March 2002 and continuing through 2003 that as UZTWM’s influence decreases, the lower zone tension water storage (LZTWM) indices increase from smaller than 0.01 to greater than 0.6 (i.e., it controls 60 percent of the model’s variance).

Figure 6.4 also shows how seasonal trends impact predictions in the summer and winter months. As an example UZTWM’s influence in the summer months shows a seasonal increasing trend until the transition to fall. This model sensitivity trend reflects the expected impact of reduced precipitation and the resultant reduction in shallow storage within both watersheds. In the winter months, the SNOW-17 parameters are only sensitive from December through April. The sensitivity of the SNOW-17 parameters is heavily influenced by the transition from dry conditions in 2001 to the wet conditions of 2003. Interestingly, the biggest difference between the SXTP1 and SPKP1 sensitivities is associated with the SNOW-17 parameters, which are maximally sensitive in the wet winter months of 2003. In general, the model’s snow response in SPKP1 is more sensitive than SXTP1, especially in dry years.

Comparison of the first order Sobol’s indices in Figure 6.4a with the total order indices in 6.4b shows that the shift from the dry conditions of 2001 to the wet conditions of 2003 increased the importance of parameter interactions on the HL-RDHM’s forecasts. Recall that the difference between a parameter’s first order and total order indices gives a measure of the importance of its interactions with other parameters. In Figure 6.4b the increased influence of parameter interactions in the wet conditions of 2003 is reflected by the increased number of parameters classified as being sensitive (designated with circles) as well as by the increases in their Sobol’s indices values relative to Figure 6.4a. For example, consider the 2002
Figure 6.4. Monthly HL-RDHM sensitivities using (a) Sobol’s first order indices and (b) Sobol’s total order indices. Sobol’s indices were computed using the RMSE measure and an hourly model time step. Triangles represent highly sensitive parameters that contribute at least 10% of the overall model output variance. Circles represent sensitive parameters that contribute at least 1% of the overall model output variance. The color legends and shading represent the Sobol indices’ magnitudes and ranges. Each row represents one month and each column represents one parameter. January and February of 2001 are missing because they represent the model warm up period.
total order indices of the maximum percolation rate (ZPERC) for both watershed models in Figure 6.4b. ZPERC shows a significant increase in the number of months where it is classified as being sensitive relative to Figure 6.4a. For the SXTP1 watershed model ZPERC’s interactions shift its classification to highly sensitive for March and October of 2002 (i.e., it explains > 10% of HL-RDHM’s output variance). These monthly results show that the transition to wet conditions increases the number parameters controlling HL-RDHM’s response as well as the importance of their interactions. Sections 6.6.3 and 6.6.4 further reduce the timescale of our analysis to flood events where it was computationally tractable to test the influence of the spatially heterogeneous forcing on the parametric sensitivities.

6.6.3 Event sensitivities based on distributed forcing and distributed parameters

Two flood events have been selected and analyzed to better understand the spatial distribution of the HL-RDHM sensitivities. This section and Section 6.6.4 focus on the SPKP1 watershed for two selected events that occurred in May 2002 and September 2003 [see Figure 6.3]. These events were selected to have a similar mass contribution and significantly different spatial rainfall distributions. The May 2002 event is fairly uniform and the September 2003 event has a heterogeneous spatial distribution. Our focus on the SPKP1 watershed was largely motivated by the extremely large computing demands posed by using Sobol’s method for spatial analysis of the HL-RDHM’s sensitivities. To further improve the tractability of this analysis the SNOW-17 component was excluded neglected since neither event was impacted by snowfall. Figure 6.3 provides a detailed spatial mapping of precipitation, first order sensitivities, and cell-level interactive sensitivities for the two events.

Figures 6.5a and 6.5d illustrate that the May 2002 and September 2003 events do represent a uniform and a heterogeneous distribution of precipitation, respectively. Overall the mean area of precipitation (MAP) of the May 2002 event is smaller than the September 2003 event. It is quite evident that the September 2003 event shown in Figure 6.5d has its most significant forcing in the northeastern boundary of the SPKP1 watershed near the outlet. The uniformly forced event
Figure 6.5. Spatial distribution of the total event precipitation and cell-level sensitivities for the SPKP1 watershed. The May 2002 event is represented by (a) its spatial precipitation distribution, (b) the first order Sobol’s indices for each model cell, and (c) the cell level interactions. The September 2003 event is represented by (d) its spatial precipitation distribution, (e) the first order Sobol’s indices for each model cell, and (f) the cell level interactions. Note cell level interactions were computed as the difference between each cell’s total order and the first order Sobol’s indices. The cell-level Sobol’s indices were computed by summing over all of individual parameter indices analyzed in each cell. The arrows in the cells designate surface flow directions.

in May 2002 yielded fairly uniform sensitivities as shown in Figures 6.5(b) and 6.5(c). After the 2 month warm up period used to model both events, the HL-RDHM’s initial conditions for the lower zone free primary water storage showed a slight increasing trend from the southwestern portion of SPKP1 (i.e., cells 1, 2, and 3) to the northeastern portion of the watershed (i.e., cells 29, 30, and 31). For the May 2002 event cells 23 and 31 had the maximum initial storage, which implies these cells were initially the wettest cells and that they should have an increased influence on the HL-RDHM’s response. Figure 6.5(b) confirms this expectation and shows that cells 23 and 31 by themselves account for more than 10-percent of the HL-RDHM’s response. Beyond the initial wetness of these cells, their close proximity to the modeled outlet for the watershed also increased their influence on predictions.
Table 6.6. Sobol’s indices for each of the 13 parameters analyzed in the May 2002 and September 2003 events. The first and total order indices for each parameter were computed by summing their individual cell-level indices over the SPKP1 watershed’s model domain.

<table>
<thead>
<tr>
<th>Event</th>
<th>Order</th>
<th>Parameters</th>
<th>Uztwm</th>
<th>Uzfwm</th>
<th>Uzk</th>
<th>Pctim</th>
<th>Adimp</th>
<th>Spere</th>
<th>Resp</th>
<th>Tzwm</th>
<th>Lzfs</th>
<th>Mlzfs</th>
<th>Lzpm</th>
<th>Lzpkm</th>
<th>Lzpke</th>
<th>Pfree</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002.5</td>
<td>1st</td>
<td>0.118</td>
<td>0.011</td>
<td>0.006</td>
<td>0.084</td>
<td>0.046</td>
<td>0.029</td>
<td>0.003</td>
<td>0.231</td>
<td>0.015</td>
<td>0.034</td>
<td>0.278</td>
<td>0.084</td>
<td>0.008</td>
<td>0.023</td>
<td>0.183</td>
</tr>
<tr>
<td></td>
<td>total</td>
<td>0.212</td>
<td>0.017</td>
<td>0.012</td>
<td>0.063</td>
<td>0.066</td>
<td>0.048</td>
<td>0.006</td>
<td>0.346</td>
<td>0.066</td>
<td>0.041</td>
<td>0.278</td>
<td>0.084</td>
<td>0.026</td>
<td>0.026</td>
<td>0.278</td>
</tr>
<tr>
<td>2003.9</td>
<td>1st</td>
<td>0.099</td>
<td>0.025</td>
<td>0.006</td>
<td>0.068</td>
<td>0.081</td>
<td>0.047</td>
<td>0.013</td>
<td>0.311</td>
<td>0.039</td>
<td>0.081</td>
<td>0.094</td>
<td>0.081</td>
<td>0.013</td>
<td>0.006</td>
<td>0.064</td>
</tr>
<tr>
<td></td>
<td>total</td>
<td>0.163</td>
<td>0.045</td>
<td>0.018</td>
<td>0.074</td>
<td>0.094</td>
<td>0.097</td>
<td>0.013</td>
<td>0.411</td>
<td>0.091</td>
<td>0.078</td>
<td>0.094</td>
<td>0.091</td>
<td>0.054</td>
<td>0.022</td>
<td>0.133</td>
</tr>
</tbody>
</table>

Overall the first order Sobol’s indices and interactivity results in Figures 6.5b and 6.5c for the May 2002 event highlight that three factors largely control the HL-RDHM’s sensitivity to a cell: (1) the cell’s initial wetness, (2) the cell’s flow connectivity (i.e., the number of arrows showing surface flow inputs), and (3) the cell’s proximity to the gauged outlet. Beyond these three factors, the spatially heterogeneous event in September 2003 also shows that significant spatial differences in forcing is a fourth factor that strongly impacts cell-level sensitivities. In Figure 6.5d, cells 24, 23, 29, 30, and 31 are very close to the SPKP1 watershed’s gauged outlet and receive significantly more rainfall than nearby cells, which strongly increases their sensitivities as shown in Figures 6.5e and 6.5f. These five cells cover only 16-percent of the modeled area but account for nearly half of the HL-RDHM’s output variance. It is interesting to note that cell 2 in the southwestern portion of the grid domain received the most rainfall overall [see Figure 6.5a], but had a relatively small impact on model predictions due to its initial conditions (i.e., it initially had a slightly increased storage capacity to capture much of the rainfall) and its significant distance from the gauged outlet.

Table 6.6 provides a summary of the first and total order Sobol’s indices for the 13 parameters sampled for the May 2002 and September 2003 events. The indices in Table 6.6 reflect the total watershed-level influence of each parameter on the event predictions. The table shows many similar trends to the prior lumped parameter results from Sections 6.6.1 and 6.6.2. For both events the model’s upper zone tension water storage (UZTWM), lower zone tension water storage (LZTWM), and the fraction of percolating water (PFREE) are the dominant parameters impacting between 16 to more than 40 percent of the HL-RDHM’s variance.

As was also seen in the annual and monthly results, the model’s upper zone tension water storage (UZTWM) and fraction of percolating water (PFREE) are more important for the relatively drier system conditions in the May 2002 event.
versus the wetter conditions of the September 2003 event. Recall from Section 6.2.2, PFREE is the fraction of water percolating from upper zone free water into lower zone free water before the the lower zone tension water is filled. PFREE is not activated when LZTWM is full which is more likely to happen during wet conditions. In both events, the lower zone tension water storage (LZTWM) is the dominant parameter explaining 35 to 41 percent of the HL-RDHM’s output variance. Figure 6.6 provides maps of the spatial distributions of these parameters’ total order indices. For the May 2002 spatially uniform precipitation event, the total order indices maps in Figures 6.6a-6.6c show each parameter has a different spatial distribution for its most important cells and a spatial trend is difficult to discern.

Alternatively, Figures 6.6d-6.6f show that the significant differences in the spatial distribution of forcing for the September 2003 event strongly influence the spatial distribution of parametric sensitivities for the HL-RDHM’s upper and lower zone tension water storage parameters (UZTWM and LZTWM). Comparison of Figures 6.6e and 6.6b confirms that the model’s fraction of percolating water (PFREE) has a reduced importance for the wet conditions of the September 2003 event versus the relatively drier conditions of the May 2002 events. In both events the lower zone tension water storage (LZTWM) was the most dominant parameter [see Table 6.6]. Readers should be careful when interpreting the maps of Figure 6.6 to note that a different color legend scale had to be used for LZTWM [i.e., Figure 6.6c] to adequately map its total order indices, which had a substantially wider range relative to the other parameters analyzed in Figure 6.6. In all of the maps of Figure 6.6 the northeastern cell 31 had a very significant impact on the HL-RDHM’s output variance. This is particularly true for the lower zone tension water storage (LZTWM) and reflects the importance of its proximity to the outlet and its initial lower zone free primary water storage (i.e., its initial wetness conditions). For the May 2002 and September 2003 events the lower zone tension water storage parameter (LZTWM) in cell 31 controlled 3 to 7-percent of the HL-RDHM’s response, respectively.

The spatial maps of Figures 6.5 and 6.6 as well as the Sobol’s indices for each parameter in Table 6.6 show that the spatial distributions of forcing and model cell wetness significantly control the HL-RDHM’s sensitivity. Section 6.6.4 seeks
to verify the Sobol’ sensitivity rankings as well as the method’s ability to screen or significantly reduce the number of parameters that must be considered when calibrating HL-RDHM.

6.6.4 Verification of event analysis sensitivity rankings

Building on Anderson [181] and Tang et al. [160], we have tested the repeatability and screening effectiveness of the Sobol’s sensitivity method. We have used the
sensitivity classifications given in Section 6.6.3 in combination with independent
LHS-based random draws to develop the verification plots for event analysis sen-
sitivity rankings given in Figure 6.7. Our overall intent for this analysis is to use
independent random samples to test if the parameters and model cells classified
as being sensitive in Section 6.6.3 do in fact control the HL-RDHM’s response.
Repeating the analysis for the May 2002 and September 2003 events also provides
some insights on how the spatial heterogeneity of forcing impacts model parameter
screening using Sobol’s method.

Our analysis uses four randomly drawn parameter sets: (1) Set 1 consists of
1000 randomly drawn Latin hypercube samples for the 13 SAC-SMA parameters
analyzed for the SPKP1 watershed model, (2) Set 2 consists of random samples
for the subset of model parameters consisting of the top 6 most sensitive SAC-
SMA parameters which are perturbed across all model cells, (3) Set 3 consists of
random samples for all 13 SAC-SMA parameters analyzed using only the top 15
most sensitive model cells, and (4) Set 4 consists of random samples for the subset
of model parameters representing the top 6 most sensitive parameters within the
top 15 most sensitive model cells. The overall rationale for using these four sets is
to simply reduce the number of model parameters (i.e., 13 parameters per cell * 31
model cells) by approximately 50 percent for Sets 2 and 3. Set 4 then combines the
parameter-based screening strategy of Set 2 with the cell-based screening strategy
of set 3 to reduce the overall number of model parameters being considered by
more than 75-percent.

In Figure 6.7, Set 1 provides the performance baseline representing the full
randomly generated independent sample set. The premise of this analysis is that
if the Sobol method’s results from Section 6.3 are correct than the method’s rank-
ings should provide sufficient information to identify the correct subset of sensitive
parameters. When the correct subset of sensitive parameters is sampled randomly
(i.e., Set 2, Set 3, or Set 4) than they should be sufficient to capture model output
from the random samples of the full parameter set in Set 1 yielding a linear
trend with an ideal correlation coefficient of 1. Several interesting observations are
evident in Figure 6.7 that depend on the type of strategy used for reducing the
number of parameters considered as well as the spatial heterogeneity of the events’
forcing.
Figure 6.7. Verification plots for event analysis sensitivity rankings based on hourly model time steps for the HL-RDHM. The scatter plots show the RMSE of streamflow predictions. Set 1 consists of 1000 randomly drawn Latin hypercube samples for the 13 SAC-SMA parameters analyzed for the SPKP1 watershed model. Set 2 consists of random samples for the subset of model parameters composed by top 6 most sensitive SAC-SMA parameters perturbed across all model cells. Set 3 consists of random samples for all 13 SAC-SMA parameters analyzed using only the top 15 most sensitive model cells. Set 4 consists of random samples for the subset of model parameters representing the top 6 most sensitive parameters on the top 15 most sensitive model cells. Term $r$ represents correlation coefficient.

The parameter-based screening strategy represented by Set 2 worked well yielding correlation coefficients of 0.873 and 0.856 for the May 2002 and September 2003 events, respectively. The Set 2 parameter-based screening strategy selected the top 6 most sensitive parameters whose total order effects explained more than 90-percent of the HL-RDHM’s output variance. For the cell-based screening strategy represented by Set 3, performance depended on the event analyzed [see Figure 6.7]. For the uniform spatial distribution of precipitation in the May 2002 event, the cell-based ranking strategy actually degraded performance for capturing the HL-RDHM’s output. This is an intuitive result since uniformly distributed precipitation did not lead to any signature spatial trends in sensitivity. Alternatively, the Set 3 cell-based strategy showed an improved screening performance for the September 2003 event since this event had signature spatial trends for precipitation and model sensitivities.

Combining the parameter screening strategies to yield Set 4 served to reduce
the overall set of parameters being analyzed to less than 100 parameters while
still maintaining correlation coefficients of 0.658 and 0.733 for the May 2002 and
September 2003 events, respectively. It is interesting to note that strong spatial
trends in precipitation appear to improve the indentifiability of the HL-RDHM-
based watershed models. Evidence to support this claim can be drawn from Figure
6.7b, which shows that the Set 4 parameter screening strategy yielded a much
higher correlation coefficient for the September 2003 event. This implies that a
much smaller set of parameters can be used to fully represent HL-RDHM’s response
for the heterogeneously forced event. Overall, the subset of parameters used in
the Set 4 screening strategy for both events explained approximately 70-percent
of HL-RDHM’s output variance. The results of Figure 6.7 confirm that Sobol’s
method provides robust sensitivity rankings and that the rankings can be used
to significantly reduce the number of parameters that must be considered when
calibrating the HL-RDHM.

6.7 Discussion and conclusions

This chapter provides a step-wise analysis of the US NWS’s distributed model-
ing framework’s (HL-RDHM) sensitivities from annual to event level time-periods
with the intent of elucidating the key parameters impacting the model’s forecasts.
This chapter demonstrates a methodology that balances the computational con-
straints posed by Sobol’s sensitivity analysis with the need to fully characterize
the HL-RDHM’s forecasting sensitivities. In the context of long-term forecasts,
HL-RDHM’s sensitivities were assessed for annual periods using 24-hour model
time steps and monthly periods using 1-hour model time steps. For the annual
and monthly analysis, the HL-RDHM’s computational demands required our use
of distributed forcing and model structure, but lumped model parameterizations
for two case study watersheds within the Juniata River basin in central Penn-
sylvania, USA. In the context of event forecasts, this chapter provides detailed
spatial analysis of the HL-RDHM’s sensitivities for two flood events simulated using
a 1-hour model time step. The events were selected to demonstrate how the
spatial heterogeneity of forcing has a significant influence on the model’s spatial
sensitivities. Our spatial analysis of event sensitivities also included an evaluation
of how well Sobol’s sensitivity method performs in identifying the principle input variables controlling the HL-RDHM’s response. Our evaluation of Sobol’s method extends the SA repeatability test recommended by Anderson [181] to a spatially distributed modeling context and demonstrates that the method provides robust sensitivity rankings and that the rankings could be used to significantly reduce the number of parameters that must be considered when calibrating the HL-RDHM.

The annual to event-level time periods analyzed in the prior section show that the HL-RDHM’s sensitivities are controlled largely by four factors: (1) variations in the model’s storage for both its upper and lower zones that occur for transitions from dry to wet conditions (or vice versa), (2) strong spatial trends in the model’s NEXRAD forcing data, (3) the model cells’ proximities to the gauged outlet where the model performance objective is computed (i.e., RMSE in this study), and (4) the cells’ flow connectivities where points of confluence have increased parametric sensitivity. It provides some insights on the types of information that should guide future forecasting methodologies. Sensitivities that arise from a cell’s “proximity to the outlet gauge” and “spatial trends in precipitation” indicate that our ability to evaluate and identify appropriate forecasting models is a function of the observation network. It is very common to adapt modeling frameworks but it is far less common to advance forecasting through adaptive design and improvement of the terrestrial and riverine observation network. The results of this chapter demonstrate that a spatially distributed simulation can be controlled by as few as 3 or 4 model cells very near a stream gauge. It is likely that increased gauging would more fully activate the spatial distribution of model sensitivities. As has been widely recognized in prior studies, the resolution, magnitude, and quality of NEXRAD observations of forcing also dominates distributed models’ responses [11, 202]. Improved gauging of both the terrestrial states predicted by the HL-RDHM as well as the atmospheric forcing driving the model appear to be important needs for advancing our ability to develop distributed forecasts.

Conceptual challenges in calibrating the HL-RDHM arise because the model exhibits a highly non-linear response behavior that is heavily dependent on the individual as well as the interactive sensitivities of its parameters. As would be expected, the number and magnitude of individual and interactive parameter sensitivities increases for the HL-RDHM when modeling wet conditions. Across all of
the time periods analyzed, the HL-RDHM’s sensitivities showed that while upper zone storage and percolation parameters are very important for dry conditions, a transition to wet conditions decreases the importance of these parameters. For wet conditions, the HL-RDHM’s lower zone storages become the dominant parameters controlling the model’s response. The annual, monthly, and event-level sensitivity trends presented in this chapter highlight that static calibration strategies that do not incorporate changing model sensitivities would likely be sub-optimal in extracting information from available data [see 31]. Operational forecasts based on the HL-RDHM would benefit from the joint use of a robust sensitivity analysis framework directly integrated into new calibration methodologies.

A clear question for operational calibration of the HL-RDHM arises in terms of what watershed properties and processes need to be carefully observed or parameterized through a priori parameters in an HL-RDHM application? Reducing the number of parameters to be calibrated means that the others would be set to reasonable values based on physical watershed properties [203]. This study has shown that larger scale watershed features such as topography and climatology heavily influence the HL-RDHM’s response (as reflected in the model’s sensitivity to cell-connectivity and forcing heterogeneity). In HL-RDHM, cell-connectivity is based the algorithm developed by Wang et al. [199] in which the topographic information contained in a high resolution DEM is aggregated to the HRAP areal cells while constraining flow directions to closely match those of the original DEM. This chapter shows that the spatial distribution of precipitation and topography can cause a very limited number of parameters over a small number of model cells to control the HL-RDHM’s response.

The emerging trend in surface hydrology towards spatially distributed simulation will require the field to embrace and advance high-performance computing to ensure these tools can be used in both scientific and operational applications [e.g., see 8, 125]. Algorithmic design and implementation for high-performance computing architectures will need to be carefully considered. In this chapter, our Sobol’s method code was implemented using a highly portable MPI parallel framework [65] so that a non-expert could utilize our computational cluster in sensitivity analysis applications. Operational use of the HL-RDHM will require computational support tools that are parallel, highly portable, and implemented in a manner that
minimizes the computational expertise needed by operational personnel.
Overview of thesis conclusions and overall contributions

This thesis has developed and comprehensively compared several tools for advancing the evaluation and identification of hydrologic models including lumped conceptual models (SAC-SMA/SNOW-17), distributed conceptual models (HL-RDHM), and semi-distributed physical hydrologic models (PIHM). These tools include evolutionary multiobjective optimization algorithms and sensitivity analysis methods, developed to exploit parallel computing. The research was partitioned into four studies.

Study 1 comprehensively assessed the relative effectiveness of state-of-the-art evolutionary multiobjective optimization tools (i.e., SPEA2, ε-NSGAII, and MOSCEM-UA) in calibrating hydrologic models. Three test cases were used to compare the algorithms’ performances. Overall, SPEA2 is an excellent benchmark algorithm for multiobjective hydrologic model calibration. SPEA2 attained competitive to superior results for most of the problems tested in study 1. ε-NSGAII appears to be superior to MOSCEM-UA and competitive with SPEA2 for hydrologic model calibration. SPEA2 fully exploits its k-means clustering diversity operator to spread solutions across the search space and more reliably escapes the false nondominated front. However, the largest challenge in using SPEA2 is effectively sizing the algorithm’s archive without prior knowledge of the true solution set. It should be noted that robust results for SPEA2 required the use of set size information to specify the algorithm’s archive size. The set size information is
automatically generated by $\varepsilon$-NSGAII with minimal user input. Trial-and-error analysis is required for maximizing the SPEA2’s performance. The biggest limitation impacting $\varepsilon$-NSGAII’s performance is related to its parent algorithm NSGAII’s diversity operator \cite{101}. It has been widely reported \cite{35, 40} that the original NSGAII converges very quickly, but its crowded tournament diversity operator can fail to promote sufficient diversity for some problems. A primary benefit of epsilon NSGAII is the algorithm’s “time continuation” feature where dynamic population sizing and random solution injection enhance solution diversity for as long is computationally feasible. $\varepsilon$-NSGAII’s ease-of-use and dynamic population sizing and archiving lead to rapid convergence to very high quality solutions with minimal user input. MOSCEM-UA’s primary strength is its estimation of the posterior parameter distributions for hydrologic model parameters. The algorithm has a limited number of parameters that need to be specified. MOSCEM-UA is however, critically sensitive to these parameters. The matrix inversion used in the algorithm’s stochastic search operators causes MOSCEM-UA’s efficiency to dramatically reduce with increases in population size and complexes. The algorithm is best suited for hydrologic model calibration applications that have small parameter sets and small model evaluation times.

Study 2 presents and comprehensively assesses EMO parallelization schemes that can help to overcome the computational limits posed by calibrating complex models and expand the range of problem domains where EMO algorithms can be applied effectively. Study 2 uses a formal metrics-based framework to demonstrate the Master-Slave (MS) and Multi-Population (MP) parallelization schemes for the $\varepsilon$-NSGAII. A key finding is that time-continuation and parallel speedups can dramatically improve the efficiency and reliability of EMO algorithms in water resources applications such as hydrologic model calibration. As stated above, time continuation is an evolutionary algorithm search enhancement that promotes solution diversity and allows the $\varepsilon$-NSGAII to maintain effective search for as long as is necessary or is computationally tractable. Although the MP version of the $\varepsilon$-NSGAII is more effective than the MS scheme when solving the artificially constructed and extremely difficult test problems such as DTLZ6, this result may wrongly bias water resources applications towards using more complicated algorithms (i.e., the MP parallelization scheme), when a simpler MS strategy may work
as well or better. The Leaf River and long-term groundwater monitoring (LTM) applications in Study 2 show that enhanced search durations and time-continuation allowed the MS version of the ε-NSGAII to rapidly and reliably improve search success rates relative to the MP parallelization scheme. The ε-NSGAII's failure rates for the Leaf River and LTM test cases result from time constraints and not from algorithmic limitations. Study 2 also demonstrates the importance of monitoring solution quality using multiple EMO performance metrics, especially when performing speedup calculations. In addition, the MS parallelization model is easy to implement and best suited for computationally intensive applications with moderate to great problem difficulties. The MP parallelization strategy is more complicated and suitable for solving extremely difficult problems with limited time constraints. Overall, the three test cases used in study 2 demonstrate that the MS and MP versions of the ε-NSGAII can broaden the size and difficulty of multiobjective water resources applications that can be solved efficiently and reliably.

The computational and conceptual challenges posed by hydrologic calibration can also be addressed by simplifying the problem using model sensitivity analysis. Study 3 tested four sensitivity analysis methods: (1) PEST, (2) RSA, (3) ANOVA, and (4) Sobol’s method on the lumped Sacramento Soil Moisture Accounting (SAC-SMA) model coupled with SNOW-17. Results from study 3 characterize model sensitivities for two medium sized watersheds within the Juniata River Basin. Comparative results presented for a 3-year time series with 1 hour, 6 hour, and 24 hour time intervals show that the fully lumped SAC-SMA/SNOW-17 model’s responses are “uniquely” determined by the performance objective specified, prediction timescale, and specific watershed being modeled. PEST and RSA both neglect parameter interactions and as a consequence yield a far less nuanced description of the models they evaluate. In a broader context, sensitivity analysis shapes the manner in which hydrologists view the processes and watershed properties impacting their model results. The basic assumptions, used in PEST and RSA, such as neglecting parameter interactions, may manifest themselves in the subsequent myriad of potential uses of the hydrologic model (e.g., flood forecasting, observation network design, reservoir management, etc.) by providing an overly simplified view of the controls on a hydrologic system. Overall ANOVA
and Sobol’s method were shown to be superior to RSA and PEST. Relative to one another, ANOVA has reduced computational requirements while Sobol’s method yielded more robust sensitivity rankings. Study 3 shows that as prediction problems in hydrology grow in complexity, our analysis techniques need to evolve to better represent and understand how models behave.

As an extension of Study 3, Study 4 characterizes the spatial and temporal variations of single parameter and multi-parameter interactions for the United States National Weather Service (US NWS)’s Hydrology Laboratory Research Distributed Hydrologic Model (HL-RDHM) using the Sobol’s global sensitivity analysis method. Study 4 seeks to carefully characterize the HL-RDHM’s sensitivities while balancing the computational demands associated with Sobol’s method using parallel computing. It provides a step-wise analysis of the HL-RDHM’s sensitivities from annual to event level time-scales with the intent of elucidating the key parameters impacting the model’s forecasts. The evaluation of Sobol’s method demonstrates that the method provides robust sensitivity rankings and that the rankings could be used to significantly reduce the number of parameters that must be considered when calibrating the HL-RDHM. The annual to event-level time periods analyzed in Study 4 show that the HL-RDHM’s sensitivities are controlled largely by four factors: (1) variations in the model’s storage for both its upper and lower zones that occur for transitions from dry to wet conditions (or vice versa), (2) strong spatial trends in the model’s NEXRAD forcing data, (3) the model cells’ proximities to the gauged outlet where the model performance objective is computed (i.e., RMSE in this study), and (4) the cells’ flow connectivities where points of confluence have increased parametric sensitivity. This suggests that static calibration strategies that do not incorporate changing model sensitivities would likely be of limited value. Operational forecasts would benefit from the joint use of a robust sensitivity analysis framework directly integrated into new calibration methodologies.

In summary, model sensitivity analysis can not only identify the most important parameters and parameter interactions but also assist in better understanding the behavior of hydrologic models. Consequently, the importance of model sensitivity analysis lies in both the fact that it can reduce model calibration complexity and that it has the potential to improve our understanding of watershed models’ per-
formance controls. Parallelization of EMO algorithms can improve their reliability and efficiency in calibrating hydrologic models automatically. This is especially important when calibrating distributed hydrologic models because of the increasing complexity of the problem as well as the severe computational burden. This thesis suggests that future extensions of this work should develop methodologies where model sensitivity should be directly incorporated into calibration dynamically for enhancing distributed hydrologic forecasting. This thesis also highlights the importance of parallel computing for advancing the evaluation and identification of hydrologic models as has been proven in studies 2 and 4. A summative list of the major contributions of this thesis is given below:

- 1st methodology for comprehensively comparing EMO tools’ effectiveness for hydrologic model calibration using a rigorous metric-based approach
- 1st study to provide effective EMO parallelization schemes that expand the overall scope of water resources problems that can be solved
- Provides a comprehensive assessment of state-of-the-art SA tools that verifies their value for watershed modeling and provides guidance in algorithm selection
- Advanced distributed watershed model evaluation and identification
  
  Global sensitivity analysis clarified the key factors controlling model performance
  
  GIS mapping of sensitivities will enhance future scientific and operational forecasting efforts
Chapter 8

Implications and future work

8.1 Implications

The thesis research has comprehensively compared, implemented, and developed sensitivity analysis and calibration tools for hydrologic models. The overall conclusions about the relative advantages and disadvantages of these tools are summarized in Chapter 7. As the results indicate, the relative performance of the tools may vary from application to application. Users should be careful to consider the assumptions, effectiveness, reliability, and interpretability of the Evolutionary Multiobjective (EMO) algorithms, sensitivity analysis tools, and parallelization strategies (e.g., the Master-Slave (MS) and Multi-Population (MP) models in Study 2). Prior to selecting and using one of these tools, it is very important the users carefully characterize the computational and conceptual challenges posed by candidate models.

Although SPEA2 was shown to be more reliable than the serial version of the $\varepsilon$-NSGAII when calibrating hydrologic models, SPEA2 requires time consuming trial-and-error analysis in order to correctly specify its archive size and population size. In addition, the code’s inconvenient structure makes it less efficient than $\varepsilon$-NSGAII. Given the competitive performance, its ease-of-use and efficient features, $\varepsilon$-NSGAII is recommended for multiobjective calibration of hydrologic models. Furthermore, the Master-Slave (MS) and Multi-Population (MP) versions of $\varepsilon$-NSGAII have broadened the type and size of the problems that can be solved. Generally, the serial $\varepsilon$-NSGAII suffices for applications with tractable
computational demand and moderate problem complexity. The computational demand is primarily determined by the model simulation time which is a function of model structure. Lumped hydrologic models have much less simulation times than distributed hydrologic models and are typically computationally tractable for a reasonable number of random seeds. The complexity of calibration problems is impacted by the number of parameters being identified, input data quality, hydrologic conditions, watershed characteristics, and model structure. Typically, the serial version of $\varepsilon$-NSGAII is sufficient for calibration of a lumped hydrologic model with about 20 parameters such as Sacramento Soil Moisture Accounting (SAC-SMA) model under normal conditions. Some special applications such as the Leaf River test cases in Study 1 and Study 2 require the MS version of $\varepsilon$-NSGAII to fully exploit "time continuation" to avoid false fronts. In most cases, the MS version of $\varepsilon$-NSGAII is desired for calibrating distributed hydrologic models with both large computation demand and significant problem complexity. The MP version is not recommended for hydrologic model calibration unless the calibration problem has proven to be extremely difficult.

Sensitivity analysis is an important tool in that it can not only identify the sensitive parameters and parametric interactions for a distributed hydrologic model but it can also help elucidate the key factors that control the model’s performance and thus provide insights about model structure. Even for lumped hydrologic models, sensitivity analysis is recommended because it can serve as a diagnostic tool to help model builders investigate how models behave based on their underlying assumptions. Based on the conclusions of Study 3, global sensitivity analysis methods accounting for parameter interactions such as ANOVA and Sobol’s method are desirable for hydrologic models. Sobol’s method is preferable to ANOVA since the method provides more robust and interpretable sensitivity results. Parallelization of Sobol’s method is necessary in order to make the global sensitivity analysis of distributed hydrologic models computationally tractable. In support of sensitivity analysis algorithms, efficient sampling schemes must be selected to generate parameter samples. The iterated fractional factorial design (IFFD) sampling scheme was found to be the best strategy for ANOVA to limit computational demands while attaining high quality results. A sample size of 10,000 for IFFD is typically sufficient for real applications. Sobol’s quasi-random sequence sampling scheme
or Latin hypercube sampling (LHS) are both suited for the Sobol’s method. The former is recommended if the parameter dimension is low ($\leq 100$) and the later is desired for high parameter dimension ($> 100$). The sample sizes are determined based on specific applications. As a rule of thumb, a sample size ranging from 2,000 to 8,000 is good for real applications. Statistical bootstrapping is necessary for both Sobol’s method and ANOVA in that it provides direct quantification of the methods’ repeatability and robustness. A resample size of 2,000 for the bootstrap is typically recommended for real applications and it has been proven in Study 3 to generate symmetric distributions of bootstrapped metrics. Combining the Sobol’s method with GIS tools provides detailed spatial distributions of sensitivity analysis results that have significant potential to enhance future scientific and operational forecasting efforts.

The SAC-SMA model is a hypothetical scientific representation of the watershed and characterizes the spatial and temporal distributions of soil moisture and hydrologic processes. The model’s ultimate goal is to simulate streamflow. The sensitivity analysis found that the storage parameters, percolation parameter, and impervious area percentages are the key parameters for the SAC-SMA model. With the transition from dry to wet conditions, there is clear shift from upper zone storages and percolation parameter to lower zone storages and temporal impervious area in parametric sensitivities. In addition, more parametric interactions were found during larger precipitation periods. The Hydrology Laboratory Research Distributed Hydrologic Model (HL-RDHM) uses the SAC-SMA model as the model for each cell in a $4km \times 4km$ rectangular grid. The model’s behavior is controlled by the storage variations, spatial trends in forcing, cell-connectivity, and cell proximity to the gauged outlet. The key findings about SAC-SMA and HL-RDHM are closely related to the structures of the models and the use of stream flow observation data in the model performance metrics. It is important to keep this in mind when analyzing results of this thesis.

8.2 Future work

In study 1, a comprehensive comparative study was conducted to elucidate the relative performance of three EMO algorithms for calibration of a fully lumped
conceptual models (SAC-SMA) and a semi-distributed version of a physical model termed Penn State Integrated Hydrologic Model (PIHM). Study 1 demonstrates that complex models such as PIHM will pose significant challenges to model calibration algorithms in terms of their computational demands and high-dimensional search spaces. Additionally, Study 1 highlights that all of the EMO algorithms experienced some failures in the hydrologic calibration problems. Therefore, Study 2 presents and comprehensively assesses EMO parallelization schemes that can help to overcome the computational limits posed by calibrating complex models and expand the range of problem domains where EMO algorithms can be applied effectively. My overall thesis conclusion is that the MS version of ε-NSGAII is more efficient and reliable in hydrologic model calibration but the MP version is more effective in solving extremely difficult problem without computational limits. Future work should be conducted to take the advantages of both versions of ε-NSGAII to develop a hybrid parallelization model. In the new model, users could specify a master processor on which multiple populations are generated and evolved. Each population on the master processor employs multiple slave processors to evaluate objective functions. After some period of evolving based on specific criteria, multiple populations will share their solutions in order to enhance diversity. Because multiple populations are on a single processor, the solution sharing does not require communication through network so that it would limit communication costs and enhance the algorithm’s scalability.

In Study 3 and Study 4, sensitivity analysis was conducted in order to reduce the parametric dimensionality for hydrologic model calibration problems. Lumped model sensitivity analysis in Study 3 reveals that model parameter sensitivities are impacted by forcing as well as initial conditions. Distributed model sensitivity analysis in Study 4 finds that storage variations, spatial trends in forcing, cell-connectivity, and cell proximity to the gauged outlet are the four primary factors that control the HL-RDHM’s behavior. These conclusions need to be further verified by conducting a synthetic study using artificially constructed data where these hypotheses can be tested and falsified explicitly.

The conclusions of Study 4 highlights that sensitivities are a function of a model cell’s proximity to the outlet gauge. However, the distributed model sensitivity analysis implemented in Study 4 only uses root mean square error (RMSE)
of the streamflow at the outlet of the watershed as model performance measure. This limits our knowledge of how the individual parameter sensitivities as well as parameter interactions might be impacted by internal watershed points. Preferably, data at multiple observation locations within the watershed should be used to identify model sensitivities to different model outputs distributed in space.

The results of Study 4 demonstrate that a spatially distributed simulation can be strongly influenced by a small number of model cells very near a stream gauge. It suggests that future extensions of the sensitivity analysis and model calibration work presented in this thesis should explore parameter grouping strategies based on the forecasted spatial distribution of forcing and flood events, distance to the gaged point, and large-scale watershed characteristics (e.g., topography, climatology, etc.) which may potentially impact model performance significantly. It is also possible to group parameters based on their relative sensitivities. For example, in Study 4, upper zone free water lateral depletion rate (UZK) is relatively insensitive and could be treated as a single spatially lumped parameter by assigning the same parameter value in all model cells. Parameter grouping is a crucial strategy because it not only reduces the complexity of the model identification problem but it also decreases the time needed for either sensitivity analysis or model calibration. Recall that in Study 4, we assumed that model parameters are lumped for long-term analysis and let parameters be distributed only for event analysis in order to make the problems computationally tractable. Through parameter grouping, it would improve our ability to analyze distributed parameter sensitivities for long-term hydrologic forecasts. Further research should also consider using dimensional analysis to group parameters and reduce the dimension of calibration or sensitivity analysis applications.

Study 3 concludes that model parameter sensitivities are heavily impacted by the choice of analysis method, model time interval, and local physical characteristics. In addition, Study 4 shows the number and magnitude of individual and interactive parameter sensitivities depend on time periods, forcing, initial conditions, and watershed topology. These studies highlight that static calibration strategies that do not incorporate changing model sensitivities would likely be sub-optimal in extracting information from available data. Future work should also focus on establishing adaptive calibration schemes which only calibrate sensitive parameters
identified using sensitivity analysis. Linking sensitivity analysis and calibration seamlessly is desirable but challenging. It will be important for multiobjective search and sensitivity analysis to be combined in a manner that reduces the computational burden of model identification. Clearly, the computational constraints posed by adaptive calibration strategies will have to be overcome by using parallel computation. Parallelization can be implemented for sensitivity analysis schemes, calibration algorithms such as evolutionary multiobjective optimization (EMO) algorithms, and the hydrologic models themselves.

Further study should be conducted to extend the sensitivity analysis and model calibration tools developed in this thesis to the fully-distributed version of PIHM. Since PIHM is an integrated model which couples processes involving interception, surface runoff, unsaturated groundwater flow, saturated groundwater flow, streamflow, and sediment transport, it is desirable to do model sensitivity analysis and calibration based on the model outputs corresponding to different processes. The sensitivity analysis approaches and calibration algorithms developed in this research are general tools which are suitable for PIHM as well as a variety of other models.
Appendix A

Source codes for sensitivity analysis tools

In this appendix, the source codes used in this thesis including Latin Hypercube Sampling (LHS), Iterated Fractional Factorial Design (IFFD), Sobol’s random sequence sampling, ANOVA calculation, Sobol’s method are presented. The source codes are written in C language.

A.1 Public functions

In this section, the public functions that are called by the sampling schemes or sensitivity analysis tools are included. The text between /* and */ signs are the comments about the functions.

/*This function allocates memory for an 1-D integer type pointer*/
int *IntVector(int n) {
    int *v;
    v=(int *) calloc( (size_t) n, (size_t) sizeof(int));
    return v;
}

/*This function frees memory for an 1-D integer type pointer*/
int FreeIntVector(int *v)
{
    free((char *) v);
    return 0;
}
/* This function allocates memory for an 2-D integer type pointer*/
int **IntMatrix(int m, int n)
{
    register int i;
    int **x;

    x=(int **) calloc((size_t) m, (size_t) sizeof(int *));
    for(i=0;i<m;i++) {
        x[i]=(int *) calloc((size_t) n, (size_t) sizeof(int));
    }
    return x;
}

/* This function frees memory for an 2-D integer type pointer*/
int FreeIntMatrix (int **x, int m)
{
    register int i;

    for (i=0;i<m;i++) free((char *) x[i]);
    free((char *) x);
    return 0;
}

/* This function generates a double type random number within the
range of [lb,ub]. RAND_MAX is the maximum value that can be returned
by the rand function. It is defined as #define RAND_MAX 0x7fff */
double RandomDblRge(double lb, double ub) {
    double x;
    x=lb+(ub-lb)/(RAND_MAX)*(double) rand();
    return (x);
}

/* This function generates a double type random number within the
range of [0,1].*/
double RandomDbl() {
    double x;
    x=1.0/(RAND_MAX)*(double) rand();
    return (x);
}

/*This function random select n numbers from 1,...,m, m>=n, without
replacement*/
void permutate(int *P, int m, int n) {
    int i,j,k;
    int *index1, *index2;
    int element, nPos;

    index1 = IntVector(m);
    index2 = IntVector(m);
for (i=0; i<m; i++)
{
    index1[i] = 1;
    index2[i] = 0;
}
for(i=0;i<n;i++)
{
    nPos = 0;
    for (j=0; j<m; j++)
        if (index1[j] > 0)
        {
            index2[nPos] = (j+1);
            nPos = nPos+1;
        }
    k = -1;
    while (k < 1) k = (int)Round(0.5+nPos*RandomDbl());
    element = index2[k-1];
    index1[element-1] = 0;
    P[i] = element;
}
FreeIntVector(index1);
FreeIntVector(index2);

A.2 Latin Hypercube Sampling

/*Latin Hypercube Sampling function
nrep-number of replicates, repeat LHS process nrep times
nSpl-number of samples at each replicate, also define the number of levels
nparm-number of parameters
lb,ub-lower and upper bound of parameter values */
void LHS(int nrep,int nSpl, int nparm, double *lb, double *ub,double **data) {

    int i,j,k;
    int *P;
    double l,u,interval;
    P=IntVector(nSpl);
    for(i=0;i<nrep;i++)
    {
        for(j=0;j<nparm;j++)
        {
            permutate(P,nSpl,nSpl);
            interval=(ub[j]-lb[j])/nSpl;
            for(k=0;k<nSpl;k++)
            {
            }
A.3 Iterated Fractional Factorial Design

/*Iterated Fractional Factorial Design function
 nrep-number of replicates, repeat LHS process nrep times
 ncol-size of Hadamard matrix, define the number of levels, must be of the form 2^k*p
 for k=1, 12 or 20
 nzerno-number of replicates that take middle level, nzero<=nrep
 npar-number of parameters
 lb,ub-lower and upper bound of parameter values
 LL is the factor level matrix, can be used in ANOVA */

\[ \begin{align*}
\{ & l = l + (P[k] - 1) \times \text{interval;} \\
& u = l + \text{interval}; \\
& \text{data}[i + nSpl + k][j] = \text{RandomDblRge}(l, u); \\
\} \\
\}
\]
for(j=0;j<nparm;j++)
{
    midwid=nzero/((double)nrep)*=(ub[j]-lb[j]);
    if(group[i][j]==0)
    {
        
        1=(lb[j]+ub[j])/2-midwid/2;
        u=(lb[j]+ub[j])/2+midwid/2;
        for(k=0;k<nrow;k++)
        {
            data[i*nrow+k][j]=RandomDblRge(l,u);
            LL[i*nrow+k][j]=0;
        }
    }
    else if(group[i][j]>0)
    {
        for(k=0;k<nrow;k++)
        {
            Levels[k]=H2[k][group[i][j]-1];
        }
        //generate sampels
        for(k=0;k<nrow;k++)
        {
            if(Levels[k]==0)
            {
                1=lb[j];
                u=(lb[j]+ub[j])/2-midwid/2;
                LL[i*nrow+k][j]=-1;
            }
            else if(Levels[k]==1)
            {
                1=(lb[j]+ub[j])/2+midwid/2;
                u=ub[j];
                LL[i*nrow+k][j]=1;
            }
            data[i*nrow+k][j]=RandomDblRge(l,u);
        }
    }
    else
    {
        for(k=0;k<nrow;k++)
        {
            Levels[k]=1-H2[k][abs(group[i][j])-1];
        }
        //generate sampels
        for(k=0;k<nrow;k++)
        {
            if(Levels[k]==0)
            {
                1=lb[j];
            }
        }
    }
\[ u = \frac{(lb[j] + ub[j])}{2} - \text{midwid}/2; \]
\[ LL[i*nrow+k][j] = -1; \]
else if(Levels[k] == 1) {
    \[ l = \frac{(lb[j] + ub[j])}{2} + \text{midwid}/2; \]
    \[ u = ub[j]; \]
    \[ LL[i*nrow+k][j] = 1; \]
    data[i*nrow+k][j] = RandomDblRge(l, u);
}
}

FreeIntMatrix(group, nrep);
FreeIntVector(Levels);

/*This function groups parameters, return a nrep*nparm matrix*/
void groupMap(int **group, int nrep, int nparm, int ncol, int nzero)
{
    int i, j, k, kk;
    int *P;

    P = IntVector(nzero);

    for(i=0; i<nparm; i++)
    {
        //select the zero replicates
        permutate(P, nrep, nzero);
        for(j=0; j<nzero; j++)
        {
            group[P[j]-1][i] = 0;
        }
        //assign group and orientation randomly to the remaining replicates
        for(j=0; j<nrep; j++)
        {
            kk = 0;
            for(k=0; k<nzero; k++)
            {
                if(P[k] == j+1) kk = -1;
                if(kk == 0)
                {
                    kk = 1;
                    while (kk < 1) kk = (int)Round(0.5+ncol*RandomDbl());
                    if(RandomDbl() > 0.5)
                        group[j][i] = kk;
                    else
                        group[j][i] = -kk;
                }
            }
        }
    }
}
FreeIntVector(P);
}

/*This function translates -1 and 1 in a Hadamard matrix to
the levels 0 and 1, lower, upper level*/
void FFLevels(int **H,int n)
{
    int i,j;
    for(i=0;i<n;i++)
        for(j=0;j<n;j++)
            if(H[i][j]==-1) H[i][j]=0;
}

/*This function folds samples with integer type*/
void foldIntSample(int **oldSample, int **newSample, int m, int n)
{
    int i,j;
    for(i=0;i<m;i++)
        for(j=0;j<n;j++)
            newSample[i][j]=oldSample[i][j];
    for(i=n;i<2*m;i++)
        for(j=0;j<n;j++)
            newSample[i][j]=1-oldSample[i-m][j];
}

/*This function is used to generate hadamard matrix n*n
The size n must be of the form 2^k*p for p=1, 12 or 20
The code is translated from matlab*/
void hadamard(int **H, int n)
{
    int i,j,k;
    int e,p;
    //base hadamard
    int h1=-1;
    int h12[12][12]=
    int h20[20][20]=

{1,-1,-1, 1, 1,-1,-1,-1, 1,-1, 1,-1, 1, 1, 1,-1,-1, 1},
{1,-1, 1, 1,-1,-1,-1, 1,-1, 1,-1, 1, 1, 1,-1,-1, 1, 1},
{1, 1,-1,-1,-1,-1, 1,-1, 1,-1, 1, 1, 1,-1,-1, 1, 1,-1,-1},
{1, 1,-1,-1,-1, 1,-1, 1,-1, 1, 1, 1,-1,-1, 1, 1,-1,-1, 1},
{1, 1,-1,-1, 1, 1, 1, 1,-1,-1, 1,-1, 1, 1,-1,-1,-1, 1},
{1, 1, 1, 1, 1, 1, 1,-1,-1,-1, 1, 1, 1,-1,-1,-1,-1, 1},
{1, 1, 1, 1, 1, 1,-1,-1,-1,-1, 1, 1, 1,-1,-1,-1,-1,-1},
{1, 1, 1, 1, 1, 1, 1,-1,-1,-1,-1,-1, 1, 1, 1,-1,-1,-1,-1},
{1, 1, 1, 1, 1,-1,-1,-1,-1,-1,-1,-1, 1, 1, 1,-1,-1,-1,-1},
{1, 1, 1, 1, 1, 1, 1, 1,-1,-1,-1,-1,-1,-1, 1, 1, 1,-1,-1,-1,-1,-1},
{1, 1, 1, 1, 1, 1, 1, 1, 1,-1,-1,-1,-1,-1,-1,-1, 1, 1, 1,-1,-1,-1,-1,-1,-1},
{1, 1, 1, 1, 1, 1, 1, 1, 1, 1,-1,-1,-1,-1,-1,-1,-1,-1, 1},
{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,-1,-1,-1,-1,-1,-1,-1,-1, 1},
{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,-1,-1,-1,-1,-1,-1,-1,-1, 1},
{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,-1,-1,-1,-1,-1,-1,-1,-1},
{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,-1,-1,-1,-1,-1,-1,-1,-1},
{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,-1,-1,-1,-1,-1,-1,-1,-1},
{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,-1,-1,-1,-1,-1,-1,-1,-1,-1},
{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,-1,-1,-1,-1,-1,-1,-1,-1,-1};

//find k if n = 2^k*p

  e = 0;
  while (n>1 && (n/2)*2 == n)
  {
    e++;
    n=n/2;
  }

  if (n!=1) e-=2; // except for n=2^k, need a multiple of 4
  if (e<0) n=-1; // trigger error if not a multiple of 4

  //Kronecker product construction
  if (n==1)
  {
    p=1;
    H[0][0]=h1;
    for(i=0;i<e;i++)
    {
      for(j=0;j<(int)pow(2.0,i);j++)
      {
        for(k=0;k<(int)pow(2.0,i);k++)
        {
          H[j][k+p*(int)pow(2.0,i)]-=H[j][k];
          H[j+p*(int)pow(2.0,i)][k]=H[j][k];
          H[j+p*(int)pow(2.0,i)][k+p*(int)pow(2.0,i)]+=H[j][k];
        }
      }
    }
  }
else if(n==3)
{
    p=12;
    for(j=0;j<p;j++)
        for(k=0;k<p;k++)
            H[j][k]=h12[j][k];
    for(i=0;i<e;i++)
    {
        for(j=0;j<(int)pow(2.0,i);j++)
            for(k=0;k<(int)pow(2.0,i);k++)
                {
                    H[j][k+p*(int)pow(2.0,i)]=H[j][k];
                    H[j+p*(int)pow(2.0,i)][k]=H[j][k];
                    H[j+p*(int)pow(2.0,i)][k+p*(int)pow(2.0,i)]=-H[j][k];
                }
    }
}
else if(n==5)
{
    p=20;
    for(j=0;j<p;j++)
        for(k=0;k<p;k++)
            H[j][k]=h20[j][k];
    for(i=0;i<e;i++)
    {
        for(j=0;j<(int)pow(2.0,i);j++)
            for(k=0;k<(int)pow(2.0,i);k++)
                {
                    H[j][k+p*(int)pow(2.0,i)]=H[j][k];
                    H[j+p*(int)pow(2.0,i)][k]=H[j][k];
                    H[j+p*(int)pow(2.0,i)][k+p*(int)pow(2.0,i)]=-H[j][k];
                }
    }
}
else
{
    printf("n must be 2^-e*p, for p = 1, 12, 20 ");
    exit(-1);
}
A.4 Sobol’s random sequence

/*This function generates sobol’s sequence
modified from "Numerical recipes in C, 2nd edition, page 312*/
#define MAXBIT 30
#define MAXDIM 100
void sobseq(int *n, float x[])
{
    int j,k,l;
    unsigned long i,im,ipp;
    static float fac;
    static unsigned long in,ix[MAXDIM+1],*iu[MAXBIT+1];
    static unsigned long mdeg[MAXDIM+1]={0,1,2,3,4,4,5,5,5,6,6,6,6,6,6,6,6,6,
    7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,
    8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,8,
    9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,
    9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,9,
    static unsigned long ip[MAXDIM+1]={0,0,1,1,2,1,4,2,4,7,11,13,13,16,16,19,22,25,
    1,4,7,8,14,19,21,28,31,32,37,41,42,50,55,56,59,62,
    14,21,22,38,47,49,50,52,56,67,70,84,97,103,115,122,
    8,13,16,22,25,44,47,52,55,59,62,67,74,81,82,87,91,94,103,104,109,122,
    124,137,138,143,145,152,157,167,173,176,181,182,185,191,194,199,218,220,
    227,229,230,234,236,241,244,253};
    static unsigned long iv[MAXDIM*MAXBIT+1];
    int tmp[1],imin;

    if (*n < 0) {
        //initial values for recurrence, modified by yong tang
        iv[0]=0;
        for(i=1;i<=MAXDIM;i++)
        {
            for(j=0;j<mdeg[i];j++)
            {
                permutate(tmp, j+1, 1);
                iv[j*MAXDIM+i] = (1 << tmp[0]) - 1;
            }
        }
        for (j=1,k=0;j<MAXBIT;j++,k++)
        {
            for (j=1,k<MAXDIM;k++)
            {
                if (k==MAXDIM) iu[j] = &iv[k];
                for (j=mdeg[k]+1;j<MAXBIT;j++)
                {
                    ipp=ip[k];
                    i=iu[j-mdeg[k]][k];
                    i ^= (i >> mdeg[k]);
                    for (l=mdeg[k]-1;l>=1;l--){
                        if (ipp & 1) i ^= iu[j-l][k];
                        ipp >>= 1;
                    }
                    iu[j][k]=i;
                }
            }
        }
    }
A.5 Analysis of Variance

/*This function calculates F values for main effects and 2-way interactions based on anova
objs is the array storing the output responses (objectives) for nsample
LL is the factor Level matrix, -1, 0, and +1
F is the array storing F values*/
void ANOVA(float *objs, int **LL, int nsample, int nparam, float *F, float *R2)
{
    int i,j,k,l;
    float Yi[3],Yj[3],GY;//Level mean, grand mean
    int ni[3],nj[3];//number of points at each level
    float Yij[3][3];
    int nij[3][3];
    float SSTR;//treatment sum of squares
    float SSTO;
    float SSE;//error sum of squares
    float MSTR;//treatment mean square
    float MSE;//error mean square
    float SSAB;//cross-factor sum of squares
    float SSTRAB;
    float SSEAB;
    float MSAB;
    float MSEAB;
}
float s;
int index;

s=0;
for(i=0;i<nsample;i++)
{
    s=s+objs[i];
}
GY=s/nsample;

SSTO=0;
for(i=0;i<nsample;i++)
{
    SSTO=SSTO+(objs[i]-GY)*(objs[i]-GY);
}
R2[0]=0;
R2[1]=0;
//main effects
for(i=0;i<nparam;i++)
{
    for(j=0;j<3;j++)
    {
        Yi[j]=0;
        ni[j]=0;
    }
    for(j=0;j<nsample;j++)
    {
        Yi[LL[j][i]+1]=Yi[LL[j][i]+1]+objs[j];
    }
    SSTR=0;
    for(j=0;j<3;j++)
    {
        if(ni[j])>0) Yi[j]=Yi[j]/ni[j];
        SSTR=SSTR+ni[j]*(Yi[j]-GY)*(Yi[j]-GY);
    }
    SSE=0;
    for(j=0;j<nsample;j++)
    {
        SSE=SSE+(objs[j]-Yi[LL[j][i]+1])*(objs[j]-Yi[LL[j][i]+1]);
    }
    MSTR=SSTR/(3-1);
    MSE=SSE/(nsample-3);
    if(MSE>0) F[i]=MSTR/MSE;
    R2[0]=R2[0]+SSTR/SSTO;
}
//Interactions
index=nparam;
for(i=0;i<nparam-1;i++)
for(j=0;j<3;j++)
{
    Yi[j]=0;
    ni[j]=0;
}
for(j=0;j<nsample;j++)
{
    Yi[LL[j][i]+1]=Yi[LL[j][i]+1]+objs[j];
}
for(j=0;j<3;j++)
{
    if(ni[j]>0) Yi[j]=Yi[j]/ni[j];
}
for(j=i+1;j<nparam;j++)
{
    for(k=0;k<3;k++)
    {
        Yj[k]=0;
        nj[k]=0;
    }
    for(k=0;k<nsample;k++)
    {
        Yj[LL[k][j]+1]=Yj[LL[k][j]+1]+objs[k];
        nj[LL[k][j]+1]=nj[LL[k][j]+1]+1;
    }
    for(k=0;k<3;k++)
    {
        if(nj[k]>0) Yj[k]=Yj[k]/nj[k];
    }
    for(k=0;k<3;k++)
    {
        for(l=0;l<3;l++)
        {
            Yij[k][l]=0;
            nij[k][l]=0;
        }
    }
    for(k=0;k<nsample;k++)
    {
        Yij[LL[k][i]+1][LL[k][j]+1]=Yij[LL[k][i]+1][LL[k][j]+1]+objs[k];
        nij[LL[k][i]+1][LL[k][j]+1]=nij[LL[k][i]+1][LL[k][j]+1]+1;
    }
    SSTRAB=0;
    for(k=0;k<3;k++)
    {
        for(l=0;l<3;l++)
        {
            SSTRAB+=Yij[k][l];
        }
    }
}
if(nij[k][l]>0) Yij[k][l]=Yij[k][l]/nij[k][l];
SSTRAB=SSTRAB+nij[k][l]*(Yij[k][l]-GY)*(Yij[k][l]-GY);
}
}
SSAB=0;
for(k=0;k<3;k++)
{
    for(l=0;l<3;l++)
    {
        SSAB=SSAB+nij[k][l]*(Yij[k][l]-Yi[k]-Yj[l]+GY)*
           (Yij[k][l]-Yi[k]-Yj[l]+GY);
    }
}
MSAB=SSAB/((3-1)*(3-1));
SSEAB=0;
for(k=0;k<nsample;k++)
{
    SSEAB=SSEAB+(objs[k]-Yij[LL[k][i]+1][LL[k][j]+1])*
           (objs[k]-Yij[LL[k][i]+1][LL[k][j]+1]);
}
MSEAB=SSEAB/(nsample-3*3);
if(MSEAB>0) F[index]=MSAB/MSEAB;
index=index+1;
}

A.6 Sobol’s method

/*This function calculates the 1st and total order sobol’s indices
opt: control argument, 0-1st order; 1-total order
a0-model output using param values from sample 2
a2-model output using param values from sample 1
a1-model output using analyzed param value from sample 1
    using other param values from sample 2*/
void sobols(float *a0, float *a1, float *a2, int nsample, float & S, int opt)
{
    int i;
    float EY2,V,U;
    float tmp1,tmp2,tmp3;
    float c;

    tmp1=0;
    for(i=0;i<nsample;i++)
    {
tmp1=tmp1+a0[i];
}
c=tmp1/nsample;
if(opt==0)
{
    tmp1=0;
    tmp2=0;
    tmp3=0;
    EY2=0;
    for(i=0;i<nsample;i++)
    {
        EY2=EY2+(a0[i]-c)*(a2[i]-c);
        tmp1=tmp1+(a2[i]-c)*(a2[i]-c);
        tmp2=tmp2+(a2[i]-c);
        tmp3=tmp3+(a1[i]-c)*(a2[i]-c);
    }
    EY2=EY2/nsample;
    V=tmp1/(nsample-1)-(tmp2/nsample)*(tmp2/nsample);
    U=tmp3/(nsample-1);
    if(V!=0) S=(U-EY2)/V;
}
if(opt==1)
{
    tmp1=0;
    tmp2=0;
    tmp3=0;
    for(i=0;i<nsample;i++)
    {
        tmp1=tmp1+(a0[i]-c)*(a0[i]-c);
        tmp2=tmp2+(a0[i]-c)*(a1[i]-c);
        tmp3=tmp3+(a0[i]-c);
    }
    EY2=(tmp3/nsample)*(tmp3/nsample);
    V=tmp1/(nsample-1)-EY2;
    U=tmp2/(nsample-1);
    if(V!=0)
    {
        S=(U-EY2)/V;
        S=1-S;
    }
}

/*This function computes the 2nd order Sobol's indices
a0-model output using param values from sample 2
a4-model output using param values from sample 1
a2,a3-model output using analyzed params' value from sample 1
    using other params' values from sample 2
a2-changing parameter i, a3-changing parameter j*/
ai-model output using param i's value from sample 2
using other params' values from sample 1*/

```c
void sobols2(float *a0, float *a1, float *a2, float *a3, float *a4,
          int nsample, float & S)
{
    int i;
    float EY, EY2,V,Vi,Vj,Vij;
    float tmp1,tmp2,tmp3,tmp4,tmp5;
    float c;

    tmp1=0;
    for(i=0;i<nsample;i++)
    {
        tmp1=tmp1+a0[i];
    }
    c=tmp1/nsample;

    EY=0;
    EY2=0;
    tmp1=0;
    tmp2=0;
    tmp3=0;
    tmp4=0;
    tmp5=0;
    for(i=0;i<nsample;i++)
    {
        EY=EY+(a0[i]-c)*(a4[i]-c);
        EY2=EY2+(a1[i]-c)*(a3[i]-c);
        tmp1=tmp1+(a1[i]-c)*(a1[i]-c);
        tmp2=tmp2+(a1[i]-c);
        tmp3=tmp3+(a1[i]-c)*(a2[i]-c);
        tmp4=tmp4+(a2[i]-c)*(a4[i]-c);
        tmp5=tmp5+(a3[i]-c)*(a4[i]-c);
    }
    EY2=EY2/nsample;
    EY=EY/nsample;
    V=tmp1/(nsample-1)-(tmp2/nsample)*(tmp2/nsample);
    U=tmp3/(nsample-1);
    Vij=U-EY;
    U=tmp4/(nsample-1);
    Vi=U-EY;
    U=tmp5/(nsample-1);
    Vj=U-EY;
    if(V!=0) S=(Vij-Vi-Vj)/V;
}
```


Lecture Notes in Computer Science 3410, Springer Verlag, Guanajuato, Mexico, pp. 386–398.


Vita

Yong Tang

Education

• Sept. 2003 - May 2007, Ph.D., Civil Engineering (Water Resources), Penn State University
  Ph.D. Minor, Computational Science, Penn State University

• Sept. 2000 - July 2002, M.S., Water Resources and Hydrology, Tsinghua University, China

• Sept. 1996 - July 2000, B.E., Hydraulic & Hydropower Construction Engineering, Tsinghua University, China

Work Experience

• July 2002 - August 2003, Civil Engineer, Project Manager, Beijing TDRISING Technology Ltd., China

Internship

• Summer 2001 & Summer 2000, Hangzhou ACE Electronic System Engineering Ltd, China

• Spring 2000, Three Gorges Dam Project, China

Selected Papers


• Tang, Y., Reed, P.M., Wagener, T., and van Werkhoven, K., “Comparing sensitivity analysis methods to advance lumped watershed model identification and evaluation”, Hydrology Earth System Science, In Review


• Tang, Y., Reed, P.M., and Wagener, T., “How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration?” Hydrology Earth System Science, 10, 289-307, 2006