EVALUATING AND DEVELOPING PARAMETER OPTIMIZATION AND UNCERTAINTY ANALYSIS METHODS FOR A COMPUTATIONALLY INTENSIVE DISTRIBUTED HYDROLOGICAL MODEL

A Dissertation

by

XUESONG ZHANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2008

Major Subject: Water Management and Hydrologic Sciences

EVALUATING AND DEVELOPING PARAMETER OPTIMIZATION AND UNCERTAINTY ANALYSIS METHODS FOR A COMPUTATIONALLY INTENSIVE DISTRIBUTED HYDROLOGICAL MODEL

A Dissertation

by

XUESONG ZHANG

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Raghavan Srinivasan
Committee Members,	Faming Liang
	Patricia K. Smith
	Francisco Olivera
Head of Department,	Ronald Kaiser

August 2008

Major Subject: Water Management and Hydrologic Sciences

ABSTRACT

Evaluating and Developing Parameter Optimization and Uncertainty Analysis Methods for a Computationally Intensive Distributed Hydrological Model. (August 2008)

Xuesong Zhang, B.S, Qingdao University, China; M.S., Beijing Normal University, China

Chair of Advisory Committee: Dr. Raghavan Srinivasan

This study focuses on developing and evaluating efficient and effective parameter calibration and uncertainty methods for hydrologic modeling. Five single objective optimization algorithms and six multi-objective optimization algorithms were tested for automatic parameter calibration of the SWAT model. A new multi-objective optimization method (Multi-objective Particle Swarm and Optimization & Genetic Algorithms) that combines the strengths of different optimization algorithms was proposed. Based on the evaluation of the performances of different algorithms on three test cases, the new method consistently performed better than or close to the other algorithms.

In order to save efforts of running the computationally intensive SWAT model, support vector machine (SVM) was used as a surrogate to approximate the behavior of SWAT. It was illustrated that combining SVM with Particle Swarm and Optimization can save efforts for parameter calibration of SWAT. Further, SVM was used as a surrogate to implement parameter uncertainty analysis fo SWAT. The results show that SVM helped save more than 50% of runs of the computationally intensive SWAT model

The effect of model structure on the uncertainty estimation of streamflow simulation was examined through applying SWAT and Neural Network models. The 95% uncertainty intervals estimated by SWAT only include 20% of the observed data, while

Neural Networks include more than 70%. This indicates the model structure is an important source of uncertainty of hydrologic modeling and needs to be evaluated carefully. Further exploitation of the effect of different treatments of the uncertainties of model structures on hydrologic modeling was conducted through applying four types of Bayesian Neural Networks. By considering uncertainty associated with model structure, the Bayesian Neural Networks can provide more reasonable quantification of the uncertainty of streamflow simulation. This study stresses the need for improving understanding and quantifying methods of different uncertainty sources for effective estimation of uncertainty of hydrologic simulation.

DEDICATION

This dissertation is dedicated to my parents, Zhongyi Zhang and Guiqing Peng, whose endless love and encouragement help me go through difficulties in my life.

ACKNOWLEDGEMENTS

I would like to acknowledge many people for helping me during my doctoral work. I would especially like to thank my advisor, Dr. Raghavan Srinivasan, for his invaluable guidance and encouragement throughout my doctoral work to develop independent thinking and research skills. His inspiration and patience has aided my growth as a researcher.

I am also very grateful to my dissertation committee, Dr. Faming Liang, Dr. Patricia Smith, and Dr. Francisco Olivera at TAMU, for their contribution and their good-natured support. Their critical reviews and insightful comments helped improve my dissertation.

I wish to thank Dr. Michael Van Liew at Montana Department of Environmental Quality. He generously provided part of the data that have been used in this study.

I am sincerely grateful to my family for their unconditional love and support while I pursued this research.

TABLE OF CONTENTS

CHAPTER Page

LIST OF FIGURES

Page

Page

Page

LIST OF TABLES

Page

Page

CHAPTER I

INTRODUCTION

1.1 Problem statement

1.1.1 Why use hydrologic models?

As a result of the limitations of hydrologic measurement techniques, we are not able to measure everything we would like to know about hydrological systems (Beven, 2000). Extrapolating the knowledge that we have obtained from laboratory experiments and field studies is a practical means for studying hydrologic processes happening in the real world where only limited measurements are available. Hydrologic models are usually used as an extrapolating tool to approximate the hydrologic processes and predict the evolution of important hydrologic variables like soil moisture, evapotranspiration, groundwater table, discharge and so on. With hydrologic models, the modeler can evaluate the impact of management practices and environmental change on future hydrologic response. For example, hydrologic models have been widely used for flood protection, design of engineered channels, assessing the impact of climate change, and predicting pollution incidents.

1.1.2 Error sources of hydrologic modeling

All models are wrong, and the simulated results of all models are wrong and uncertain. But some models are useful, and some are more useful than others (Marian Scott, 2004). In order to evaluate whether a model is valid and its results are useful, we need to identify the errors and uncertainty of models and their outputs. In this dissertation, "hydrologic model" is taken as an example for identifying the errors and uncertainty sources. The error and uncertainty sources of hydrologic modeling have been

---

This dissertation follows the style of *Transactions of the American Society of Agricultural and Biological Engineers*.

analyzed in previous studies (Beven, 2001; Montanari and Brath, 2004; Schaefli et al., 2006; Ewen et al., 2006). Ewen et al. (2006) gave an comprehensive description of the error sources of hydrologic modeling, which were categorized into three groups: 1) model structure error, associated with the model's equations, 2) parameter error, associated with the parameter values used in the equations; and 3) run time error, associated with rainfall and other forcing data. Further, Ewen et al. (2006) introduced the error components for each group (Table 1-1). The error components listed above all contribute to the "integrated" model output error, but their individual contribution usually cannot be isolated because the modeling process is complex and there is a lack of knowledge about the catchment and its hydrological responses (Beven, 2001; Ewen, 2006).

Table 1-1. Error components of hydrologic modeling.

M1	It is a philosophical question whether any model can exactly represent the truth, so even the best possible model might give "integrated" error.
M2	From conceptual and mathematical simplification
M3	From using approximate numerical solutions, finite time steps, etc
M4	From conceptual, mathematical and programming mistakes made by the modeler
P1	From incomplete or erroneous calibration data (i.e. forcing and response data used in calibration)
P2	From the calibration process, to compensate for model structure error
P3	From not using the optimum parameter values
P4	From mistakes made by the modeler in setting parameter values (the typing error described above contributes to component P4)
R1	From incomplete and erroneous forcing data
R2	From mistakes in forcing data made by the modeler and from mistakes in the way the model is used and the results interpreted

Note: For the model structure errors group, its components start with "M". Similarly, error components belonging to the parameter error group start with "P", and errors components belonging to the run time error group start with "R".

1.1.3 General procedures for applying a hydrologic model

When applying a hydrologic model, we would like to ask questions like (Marian Scott, 2004): Is the model valid? Are the assumptions reasonable? Does the model make sense based on the best scientific knowledge? Is the model credible? Do the model predictions match the observed data? How uncertain are results? In order to answer these questions, several key procedures followed in practical application of hydrologic model (Beven, 2000; Bedient et al., 2001) are as follows: 1) Select model based on study objectives and the perception and knowledge of the watershed under investigations. For example, if the snowfall and snowmelt processes are significant in the watershed under investigation, the snow routing component of the hydrologic model should be paid more attention to; or if the modeler would like to study an urbanized watershed, then a model developed for rural area should not be used. With the availability of observed data and the increasing need of understanding the hydrologic processes through the watershed system, physically based, distributed hydrologic models (e.g. MIKE SHE, SWAT, TOPMODEL) have been widely used to investigate water resources related problems. 2) Obtain necessary input data (e.g., precipitation, temperature, infiltration, physiography, land use, soil, channel characteristics, streamflow, ponds, and reservoirs). After the identification of a hydrologic model with specific model structure, the modelers would try their best to collect the most accurate forcing data for the hydrologic model. For example, the corrected radar rainfall data are preferable to the observed data from rain gauges for distributed hydrologic modeling; or the SSURGO soil data with high resolution (1: 24,000) would be preferable to the STATSGO soil data with resolution of 1:250,000 (Peschel et al., 2006). 3) Conduct model calibration and uncertainty analysis. Even for physically based hydrologic models, there are parameters that can not be observed directly because of the measurement limits or scaling issues. These parameters need to be estimated by parameter optimization or uncertainty analysis methods. 4) Evaluate usefulness of the model and comment on needed change or modifications. 5) Use the model to perform simulations to assess the effect of management practices, climate change, and government policies on the hydrologic system.

1.1.4 Research problems

Among the general procedures of applying hydrologic models, the selection of model structure, collecting data, and parameter calibration and uncertainty analysis are important since the accuracy of results is determined by these procedures. When applying a hydrologic model, the model structure and data available are usually fixed, while the procedure of determining parameters is relatively flexible. With highly parameterized distributed hydrologic models, parameter calibration and uncertainty analysis are a major concern of many applications of hydrologic models. Therefore, the objectives of this dissertation are to developing methods and computer tools to facilitate the robust parameter calibration and uncertainty analysis of physically based, complex hydrologic model. As the operation of these models is very time consuming, the research of this study will focus on only one of these physically based hydrologic models, the Soil and Water Assessment Tool (SWAT), that have been widely applied worldwide.

1.2 Scope of this dissertation

1.2.1 Parameter calibration for SWAT

The parameters of hydrologic model are difficult to estimate through measurement and prior estimation. Some of the conceptual model parameters can not be measured directly, such as Curve Number (CN). Some model parameters can be measured directly, such as soil hydraulic conductivity, LAI (Leaf Area Index), but suffer from experimental constraints and scaling problems (measurement scale and model scale are different). Studies have generally found that, even using intensive series of measurements of parameter values, the results have not been entirely satisfactory (Beven, 2000). In general, when modelers apply the SWAT model in a practical situation, calibration of model parameters is a necessary and critical procedure. Many studies have been conducted to find effective and efficient method for hydrologic model calibration. There are mainly two types of calibration methods: manual and automatic calibration. Traditional manual calibration is labor-intensive and subjective to modeler's opinion. Automatic methods are becoming more and more popular because of their ability to take advantage of power and speed of computers while being objective and relatively easy to

implement. The parameter calibration tool for SWAT includes popular single-objective and multi-objective evolutionary algorithms that have been successfully applied for optimizing hydrologic models or other complex systems.

1.2.2.1 Single-objective method

With the popularity of complex, physically based hydrologic model, the time consumed by running these models has increased substantially, although the speed and capacity of computers have increased multi-fold in the past several decades. Comparing and evaluating the efficacy of different optimization algorithms for calibrating the computationally intensive SWAT model is becoming a nontrivial issue. Previous studies have shown that different optimization algorithms exhibit varied performance for parameter calibration. For example, Cooper et al. (1997) evaluated Shuffled Complex Evolution algorithm (SCE-UA), Genetic Algorithm (GA) and Simulated Annealing (SA) methods for optimization of the Tank model. Chen et al. (2005) compared the performance of multi-start Powell and SCE-UA methods for calibrating the Tank model. These comparison results reveal the promising application of evolutionary algorithms. Besides the SCE-UA and GA algorithms, the Particle Swarm Optimization (PSO) has also been used to optimize the arrangement of hydraulic devices in a pipeline system (Jung and Karney, 2006), and train the Artificial Neural Networks' weights for river stage prediction (Chau, 2006). Other evolutionary algorithms, such as Differential Evaluation (DE) (Storn and Price, 1997) and Artificial Immune Systems (AIS) (de Castro and Von Zuben, 2002a; de Castro and Von Zuben, 2002b), although rarely used in hydrologic model calibration, showed promising ability for global optimization of complex systems. In this study, five global optimization algorithms (GA, SCE, PSO, DE, and AIS) will be tested for automatic parameter calibration of the SWAT model, and suggestions on selection of optimization algorithms will be provided.

1.2.2.2 Multi-objective method

Most real-world decision making problems involve multiple and conflicting objectives. In the single objective case, one attempts to obtain the best solution, which is absolutely superior to all other alternatives. In the multiple objective cases, there does

not necessarily exist a solution that is best with respect to all objectives because of incommensurability and conflict among objectives. The purpose of multi-objective algorithm is finding the Pareto optimal solutions. Several methods have been developed for multi-objective optimization of hydrologic model. For example, Yapo et al. (1998) and Gupta et al. (1998) extended SCE-UA to address multi-objective functions in the multi-objective complex evolution (MOCOM-UA) algorithm, which was further improved to a multi-objective shuffled complex evolution Metropolis (MOSCEM-UA) by Vrugt et al. (2003b). Gill et al. (2006b) developed a PSO based multi-objective algorithm, and applied it for parameter calibration of the SAC-SMA Model and Support Vector Machine. There are many methods could be used to realize this objective. Here in, six state-of-the-art algorithms (Strength Pareto Evolutionary Algorithm 2 (SPEA2), Non-dominated Sorted Genetic Algorithm II (NSGAII), Epsilon Dominance Non-dominated Sorted Genetic Algorithm II (ε-NSGAII), Multi-objective Particle Swarm Optimization algorithm (MOSPO) (Coello Coello et al., 2004), elitist-mutation MOPSO (EM-MOPSO) (Reddy and Kumar, 2007), and a variant of MOPSO develop by Gill et al. (2006b) were evaluated and compared for multi-objective parameter calibration of SWAT. After comparison of several state of the art multi-objective algorithms, the most promising single-objective optimization algorithm will be incorporated into the multi-objective optimization framework to improve the efficiency and effectiveness of the original algorithms.

1.2.2.3 Comparison between single objective and multi-objective method

In many applications of hydrologic models, single objective optimization methods have been used. For SWAT, few studies have reported the application of multi-objective optimization methods. Therefore, comparison between single objective and multi-objective methods is helpful to provide insights into the sensitivity of distributed hydrologic simulation to different calibration methods and the advantages and disadvantages of different parameter estimation methods.

1.2.3 Parameter uncertainty analysis for SWAT

Referring to Montanari and Brath (2004), the methods used to estimate the predictive uncertainty of hydrologic models were categorized into three major groups in this study. The first option is to structure the hydrologic model as a probability model, then the confidence interval of model output can be computed (Montanari et al., 1997). Representative methods of this category include Markov Chain Monte Carlo (MCMC) and a Generalized Likelihood Uncertainty Estimation (GLUE) (Beven and Binley, 1992). The second option is to analyze the statistical properties of the hydrologic model error series that occurred in reproducing observed historical river flow data. Parameter solutions (ParaSol) (van Griensven and Meixner, 2006) is a typical of this group of methods. The third option is using on-line data assimilation algorithms, like Ensemble Kalman Filter (EnKF) (Vrugt et al., 2005), Particle Filter (PF) (Moradkhani et al., 2005) and Bayesian Recursive Estimation (BaRE) (Thiemann et al., 2001). These methods have been used to estimate the uncertainty interval of real time model prediction. As SWAT is designed to evaluate long term hydrologic effect of land cover change, Climate change and management practices, the recursive methods are not applicable for SWAT model. In this study, the author would like to explore the applicability of the GLUE and MCMC methods for the analysis of predictive uncertainty of SWAT model. The effectiveness of the parameter uncertainty analysis for estimating uncertainty of hydrologic modeling will be evaluated, and the effect of taking other error sources into account will be discussed.

1.2.4 Efficient surrogate models for approximating SWAT

Function approximation is an efficient method for parameter calibration and uncertainty analysis of computationally intensive model (Gutmann, 2001). Several studies have applied different learning machines as surrogate models to approximate the behavior of computationally intensive environmental models. For example, Artificial Neural Networks (ANN) have been used by Morshed and Kaluarachchi (1998), Aly and Peralta (1999), Johnson and Rogers (2000), Almasri and Kaluarachchi (2005), and Zou et al. (2007) as surrogate of complex environmental models for parameter selection and

management practices evaluation. Radial Basis Function (RBF) has been used by Gutmann (2001), Regis and Shoemaker (2005), and Mugunthan and Shoemaker (2006) as an approximation tool of computationally intensive model for parameter calibration and uncertainty analysis. It was shown in previous studies that different learning machines exhibited various ability to approximate different models' behavior. In this study, the performances of ANN and SVM will be evaluated for approximating the SWAT model. Several practical issues related to how to efficiently and effectively apply the learning machines were also analyzed and discussed. In order to save huge amount of efforts running the computationally intensive SWAT model, the promising learning machines will be incorporated into the GLUE and MCMC methods. The efficiency and usefulness of the surrogate learning machines will be exhibited in experimental watersheds.

1.3 Dissertation organization

The dissertation consists of several chapters that deal with different aspects of the research scope. Specific topics for each individual chapter are summarized as follows:

1) Chapter II introduces the SWAT model and study area;

2) Chapter III evaluates and compares the effectiveness and efficiency of different single-objective evolutionary optimization algorithms for parameter calibration of SWAT;

3) Chapter IV evaluates the advantage and disadvantage of single-objective and multi-objective optimization algorithms;

4) Chapter V attempts to develop and compare different multi-objective evolutionary optimization algorithms for effective and efficiency parameter calibration of SWAT;

5) Chapter VI aims to approximate the behavior of the SWAT model using learning machines, which are in turn used as surrogate for efficient parameter calibration and uncertainty analysis of SWAT;

6) Chapter VII conducts parameter uncertainty analysis of SWAT using Generalized Likelihood Uncertainty Estimation (GLUE) and Bayesian Markov Chain Monte Carlo

(MCMC) methods, and evaluate the applicability of learning machines for saving efforts of running the computationally intensive SWAT model;

7) Chapter VIII discusses the effect of model structure on the uncertainty estimation of hydrologic modeling;

8) Chapter IX provides the conclusions.

CHAPTER II

SWAT MODEL AND STUDY AREA DESCRIPTION

2.1 SWAT developmental history and overview

SWAT is a continuous, long-term, distributed-parameter model designed to predict the impact of land management practices on the hydrology, sediment and contaminant transport in agricultural watersheds (Arnold et al., 1998). SWAT model was developed through incorporating features of several ARS models, including the SWRRB (Simulator for Water Resources in Rural Basins), CREAMS (Chemicals, Runoff, and Erosion from Agricultural Management Systems), GLEAMS (Groundwater Loading Effects on Agricultural Management Systems), and EPIC (Erosion-Productivity Impact Calculator). Gassman et al. (2007) presented the developmental history of SWAT in a schematic chart (Figure 2-1).

Development of SWRRB began with modification of the daily rainfall hydrology model from CREAMS. The SWRRB was further improved through a) incorporation of the GLEAMS pesticide fate component; b) optional SCS technology for estimating peak runoff rates; and c) newly developed sediment yield equations. Also ROTO (Routing Outputs to Outlet) (Arnold et al., 1995) was linked with SWRBB to provided a reach routing approach and overcame the SWRRB subbasin limitation by "linking" multiple SWRRB runs together. Further ROTO and SWRRB were merged into a single model - SWAT, which retained all the features of SWRRB while allowing simulation of very extensive areas through subdividing it into thousands of subbasins.

Since the appearance of SWAT in the early 1990s, it has undergone continued review and expansion of capabilities (Neitsch et al., 2005a), and different versions of SWAT were developed. In this study, the newly developed SWAT2005 was applied, so

a brief overview of this version of SWAT is introduced. For a detailed description of SWAT, please refer to Neitsch et al. (2005a; 2005b) and Gassman et al. (2007).

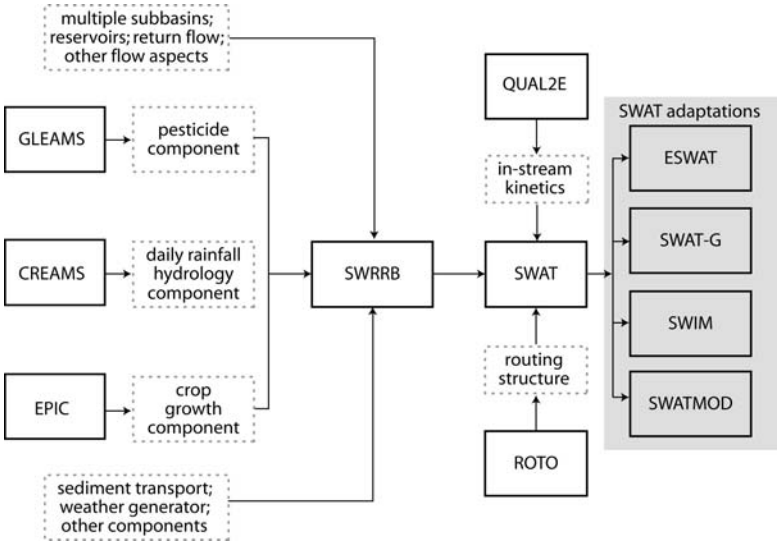

Figure 2-1. Schematic of SWAT developmental history, including selected SWAT adaptations (From Gassman et al., 2007).

2.2 SWAT model structure overview

SWAT is a basin-scale, continuous-time model that operates on a daily time step and is designed to predict the impact of management on water, sediment, and agricultural chemical yields in ungauged watersheds (Arnold et al., 1998). SWAT subdivides a watershed into sub-basins connected by a stream network, and further delineates HRUs (hydrologic response unit) consisting of unique combinations of land cover and soils within each sub-basin. The model assumes that there are no interactions among HRUs, and these HRUs are virtually located within each sub-basin. HRU delineation minimizes the computational costs of simulations by lumping similar soil and land use areas into a single unit (Neitsch et al., 2005a). HRU represent percentages of the subwatershed area and are not identified spatially within a SWAT simulation. Alternatively, a watershed can be subdivided into only subwatersheds that are characterized by dominant land use,

soil type, and management (Gassman et al., 2007). GIS based interfaces have been developed to facilitate preparing of spatial data for SWAT (e.g. Di luzio et al., 2004; Olivera, 2006).

SWAT is able to simulate surface and subsurface flow, sediment generation and deposit, and nutrient fate and movement through the landscape and river. In this chapter, only the hydrologic components of SWAT will be described. The hydrologic routines within SWAT account for snow accumulation and melt, vadose zone processes (i.e., infiltration, evaporation, plant uptake, lateral flows, and percolation), and groundwater flows. Surface runoff volume is estimated using a modified version of the USDA-SCS curve number method (USDA-SCS, 1972). A kinematic storage model (Sloan et al., 1983) is used to predict lateral flow, whereas return flow is simulated by creating a shallow aquifer (Arnold et al., 1998). Channel flood routing is estimated using the Muskingum method. Outflow from a channel is also adjusted for transmission losses, evaporation, diversions, and return flow.

2.3 Hydrologic components of SWAT

SWAT allows a number of different physical processes to be simulated in a watershed. The physical processes simulated in SWAT can be divided into two phases: land and routing.

2.3.1 Land phase of the hydrologic cycle

The hydrologic cycle as simulated by SWAT is based on the water balance equation:

$$SW_t = SW_0 + \sum_{i=1}^{t}(R_{day} - Q_{surf} - E_a - w_{seep} - Q_{gw}) \qquad \textbf{2- 1}$$

where SW_t is the final soil water content (mm H_2O), SW_0 is the initial soil water content on day i (mm H_2O), t is the time (days), R_{day} is the amount of precipitation on day i (mm H_2O), Q_{surf} is the amount of surface runoff on day i (mm H_2O), E_a is the amount of evapotranspiration on day i (mm H_2O), w_{seep} is the amount of water entering the vadose zone from the soil profile on day i (mm H_2O), and Q_{gw} is the amount of return flow on day i (mm H_2O). Figure 2-2 shows the Land phase hydrologic processes modeled by SWAT.

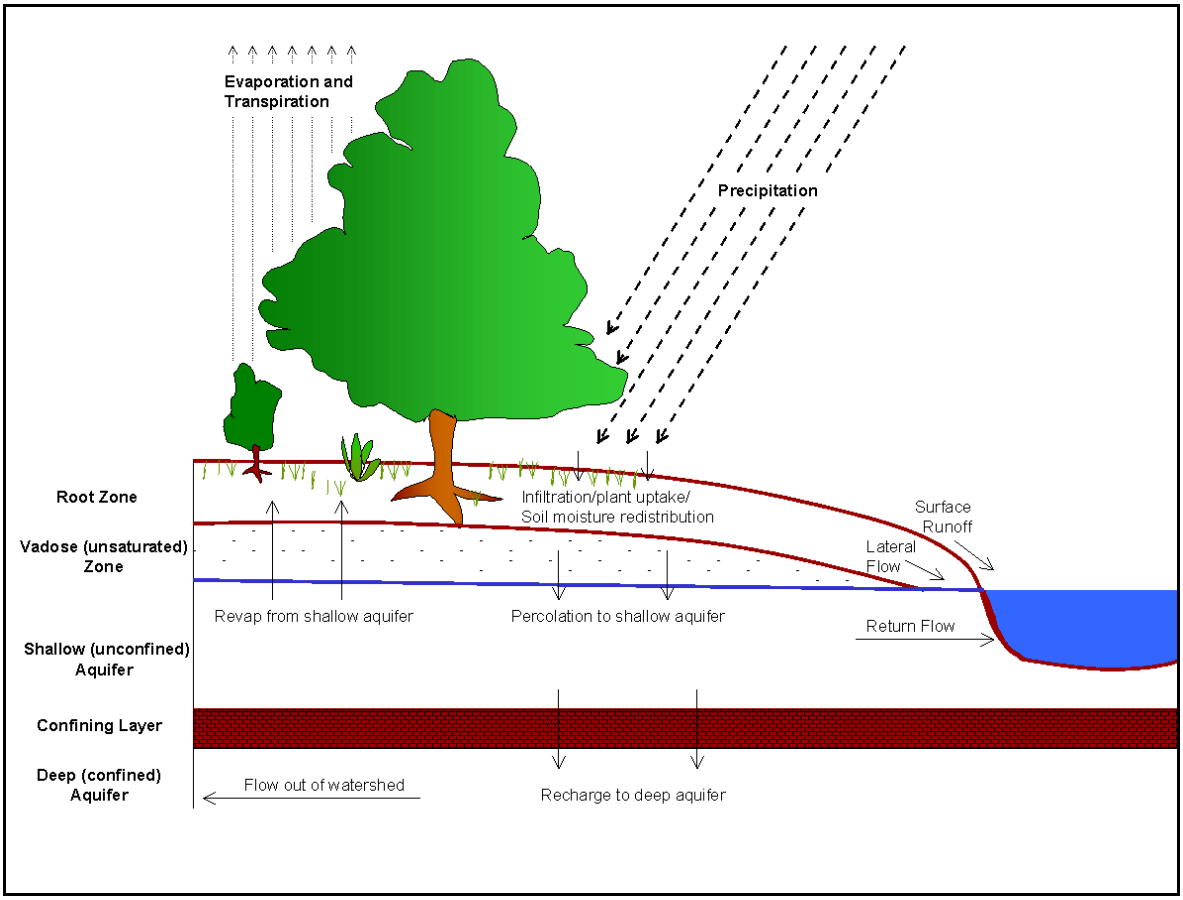

Figure 2-2. Land phase hydrologic processes modeled by SWAT (From Neitsch et al., 2005a).

2.3.1.1 Climate

The climate of a watershed provides the moisture and energy inputs that control the water balance and determine the relative importance of the different components of the hydrologic cycle. The climatic variables required by SWAT consist of daily precipitation, maximum/minimum air temperature, solar radiation, wind speed and relative humidity.

2.3.1.2 Hydrology

Precipitation may be intercepted and held in the vegetation canopy or fall to the soil surface. Water on the soil surface will infiltrate into the soil profile or flow overland as runoff. Runoff moves relatively quickly toward a stream channel and contributes to

short-term stream response. Infiltrated water may be held in the soil and later evapotranspired or it may slowly make its way to the surface-water system via underground paths. The potential pathways of water movement simulated by SWAT in a HRU are illustrated in Figure 2-2.

Canopy storage. SWAT contains use two methods to calculate the water intercepted by vegetative surfaces (the canopy) where it is held and made available for evaporation. When using the curve number method to compute surface runoff, canopy storage is taken into account in the surface runoff calculations. However, if methods such as Green & Ampt are used to model infiltration and runoff, canopy storage must be modeled separately.

Infiltration. Infiltration refers to the entry of water into a soil profile from the soil surface. As infiltration continues, the soil becomes increasingly wet, causing the rate of infiltration to decrease with time until it reaches a steady value. The initial rate of infiltration depends on the moisture content of the soil prior to the introduction of water at the soil surface. The final rate of infiltration is equivalent to the saturated hydraulic conductivity of the soil. Because the curve number method used to calculate surface runoff operates on a daily time-step, it is unable to directly model infiltration. The amount of water entering the soil profile is calculated as the difference between the amount of rainfall and the amount of surface runoff.

Redistribution. Redistribution refers to the continued movement of water through a soil profile after input of water (via precipitation or irrigation) has ceased at the soil surface. Redistribution is caused by differences in water content in the profile. Once the water content throughout the entire profile is uniform, redistribution will cease. The redistribution component of SWAT uses a storage routing technique to predict flow through each soil layer in the root zone. Downward flow, or percolation, occurs when field capacity of a soil layer is exceeded and the layer below is not saturated. The flow rate is governed by the saturated conductivity of the soil layer. Redistribution is affected by soil temperature. If the temperature in a particular layer is 0°C or below, no redistribution is allowed from that layer.

Evapotranspiration. Evapotranspiration is a collective term for all processes by which water in the liquid or solid phase at or near the earth's surface becomes atmospheric water vapor. Potential soil water evaporation is estimated as a function of potential evapotranspiration and leaf area index (area of plant leaves relative to the area of the HRU). The model offers three options for estimating potential evapotranspiration: Hargreaves, Priestley-Taylor, and Penman-Monteith (Neitsch et al., 2005a). Actual soil water evaporation is estimated by using exponential functions of soil depth and water content. Plant transpiration is simulated as a linear function of potential evapotranspiration and leaf area index.

Lateral subsurface flow. Lateral subsurface flow, or interflow, is streamflow contribution which originates below the surface but above the zone where rocks are saturated with water. Lateral subsurface flow in the soil profile (0-2m) is calculated simultaneously with redistribution. A kinematic storage model is used to predict lateral flow in each soil layer. The model accounts for variation in conductivity, slope and soil water content.

Surface runoff. Surface runoff, or overland flow, is flow that occurs along a sloping surface. Using daily or subdaily rainfall amounts, SWAT simulates surface runoff volumes and peak runoff rates for each HRU. As the SCS curve number equation is closely related to the many important hydrologic processes (e.g., vegetation interception, infiltration, soil water redistribution, and surface runoff), it is introduced as follows (Neitach et al., 2005a*)*

$$Q_{surf} = \frac{\left(R_{day} - I_a\right)^2}{\left(R_{day} - I_a + S\right)} \qquad \textbf{2- 2}$$

where Q_{surf} is the accumulated runoff or rainfall excess (mm H_2O), R_{day} is the rainfall depth for the day (mm H_2O), I_a is the initial abstractions which includes surface storage, interception and infiltration prior to runoff (mm H_2O), and S is the retention parameter (mm H_2O). The retention parameter varies spatially due to changes in soils, land use, management and slope and temporally due to changes in soil water content. The retention parameter is defined as:

$$S = 25.4\left(\frac{1000}{CN} - 10\right) \quad \textbf{2- 3}$$

where CN is the curve number for the day. The initial abstractions, I_a, is commonly approximated as $0.2S$ and equation 2-2 becomes

$$Q_{surf} = \frac{\left(R_{day} - 0.2S\right)^2}{\left(R_{day} + 0.8S\right)} \quad \textbf{2- 4}$$

Runoff will only occur when $R_{day} > I_a$.

Ponds. Ponds are water storage structures located within a subbasin which intercept surface runoff. The catchment area of a pond is defined as a fraction of the total area of the subbasin. Ponds are assumed to be located off the main channel in a subbasin and will never receive water from upstream subbasins. Pond water storage is a function of pond capacity, daily inflows and outflows, seepage and evaporation. Required inputs are the storage capacity and surface area of the pond when filled to capacity. Surface area below capacity is estimated as a non-linear function of storage.

Tributary channels. Two types of channels are defined within a subbasin: the main channel and tributary channels. Tributary channels are minor or lower order channels branching off the main channel within the subbasin. Each tributary channel within a subbasin drains only a portion of the subbasin and does not receive groundwater contribution to its flow. All flow in the tributary channels is released and routed through the main channel of the subbasin. SWAT uses the attributes of tributary channels to determine the time of concentration for the subbasin. Water losses from the channel are a function of channel width and length and flow duration. Both runoff volume and peak rate are adjusted when transmission losses occur in tributary channels.

Return flow. Return flow, or base flow, is the volume of streamflow originating from groundwater. SWAT partitions groundwater into two aquifer systems: a shallow, unconfined aquifer which contributes return flow to streams within the watershed and a deep, confined aquifer which contributes return flow to streams outside the watershed (Arnold et al., 1993). Water percolating past the bottom of the root zone is partitioned into two fractions—each fraction becomes recharge for one of the aquifers. In addition

to return flow, water stored in the shallow aquifer may replenish moisture in the soil profile in very dry conditions or be directly removed by plant. Water in the shallow or deep aquifer may be removed by pumping.

2.3.2 Routing phase of the hydrologic cycle

Once SWAT determines the loadings of water, sediment, nutrients and pesticides to the main channel, the loadings are routed through the stream network of the watershed using a command structure similar to that of HYMO (Williams and Hann, 1972). In addition to keeping track of mass flow in the channel, SWAT models the transformation of chemicals in the stream and streambed.

As water flows downstream, a portion may be lost due to evaporation and transmission through the bed of the channel. Another potential loss is removal of water from the channel for agricultural or human use. Flow may be supplemented by the fall of rain directly on the channel and/or addition of water from point source discharges. Flow is routed through the channel using the Muskingum routing method.

2.4 Review of parameter calibration and uncertainty analysis of SWAT

Numerous sensitivity analysis approaches have been reported in the SWAT literature, which provide valuable insights regarding which input parameters have the greatest impact on SWAT output. As previously discussed, the vast majority of SWAT applications report some type of calibration effort. SWAT input parameters are physically based and are allowed to vary within a realistic uncertainty range during calibration. Sensitivity analysis and calibration techniques are generally referred to as either manual or automated, and can be evaluated with a wide range of graphical and/or statistical procedures.

Table 2-1.Parameters for calibration in SWAT model

Parameter Code	Parameter	Description	Range
Parameters governing surface water response			
1	CN2	Curve Number	±20%
2	ESCO	Soil Evaporation compensation factor	0 to 1
3	SOL_AWC	Available soil water capacity	±20%
Parameters governing subsurface water response			
4	GW_REVAP	Ground water reevaporation coefficient	0.02 to 0.2
5	REVAPMN	Threshold depth of water in the shallow aquifer for reevaporation to occur (mm).	0-500
6	GWQMN	Threshold depth of water in the shallow aquifer required for return flow to occur (mm)	0-5000
7	GW_DELAY	Groundwater delay (days)	0-50
8	ALPHA_BF	Base flow recession constant	0 to 1
9	RCHRG_DP	Deep aquifer percolation fraction	0-1
Parameters governing basin response			
10	CH_K2	Effective hydraulic conductivity in main channel alluvium (mm/hr)	-0.01-150
11	TIMP	Snow pack temperature lag factor	0-1
12	SURLAG	Surface runoff lag coefficient (day)	0 to 10
13	SFTMP	Snow melt base temperature (ºC)	0-5
14	SMTMP	Snowfall temperature (ºC)	0-5
15	SMFMX	Maximum snowmelt factor for June 21 (mm H2O/ºC-day)	0 to 10
16	SMFMN	Minimum snowmelt factor for Dec. 21 (mm H2O/ºC-day)	0 to 10

One important factor that impacts the complexity of the optimization problem is the parameter dimension that need to be adjusted and the parameters' ranges. van Griensven et al. (2006) conducted detailed global sensitivity analysis of the parameters in SWAT, and the results showed that ten parameters are sensitive to the hydrologic simulation of

SWAT. Van Liew et al. (2007) tested the suitability of SWAT for the CEAP (Conservation Effect Assessment Project) in five USDA agricultural research service watersheds. In the study conducted by Van Liew et al. (2007), sixteen parameters, which include the ten parameters identified by van Griensven et al. (2006), were adjusted to calibrate the SWAT model for hydrologic simulation. In order to be consistent with previous work, the 16 parameters identified by Van Liew et al. (2007) were applied in this study. The general description of the sixteen parameters is shown in Table 2-1. The parameters' ranges were determined according to van Griensven et al. (2006) and Neitsch et al. (2005b). Among these sixteen parameters, nine of them govern surface and subsurface water response in SWAT, and other seven parameters govern basin response.

2.4.1 Sensitivity analysis of SWAT

Sensitivity analysis has been taken as an attractive method for reducing parameter dimension. Numerous studies have reported the parameter sensitivity analysis of SWAT. The sensitivity analysis methods can be categorized as either manual or automatic. Although manual methods were used to determine sensitive parameters of SWAT (e.g., Spruill et al., 2000; Lenhart et al., 2002), the automatic methods became more popular with the availability of powerful computer. Francos et al. (2003) demonstrated an application of a two-step sensitivity analysis approach for SWAT in the 3,500 km^2 Ouse Watershed in the United Kingdom using 82 input and 22 output parameters. They applied a "Morris" screening procedure that is based on the One factor At a Time (OAT) design to determine the qualitative ranking of an entire input parameter set for different model outputs at low computational cost, and then used a Fourier Amplitude Sensitivity Test (FAST) method to provide an assessment of the most relevant input parameters for a specific set of model output. Holvoet et al. (2005) combined the Latin Hypercube (LH) with OAT to determine which of 27 SWAT hydrologic-related input parameters are most sensitive regarding streamflow and atrazine outputs were for 32 km^2 Nil Watershed in central Belgium. Similar LH-OAT method was used by van Giensvcn et al. (2006) to assess the sensitivity of 41 input parameters on SWAT flow, sediment, total N, and total P estimates for both the 933 km^2 Upper North Bosque River Watershed (UNBRW)

located in Central Texas and the 3240 km^2 Sandusky River Watershed in Ohio. The LH-OAT method has been incorporated as part of the automatic sensitivity/calibration package included in SWAT2005.

2.4.2 Parameter calibration approaches for SWAT

Manual and automatic parameter calibration techniques have been widely used to estimate optimum parameters for SWAT. The manual calibration method is to utilize knowledge of the watershed and experience with the model to adjust the parameters through a trial and error procedure, which is subjective and labor intensive (Gupta et al., 1999). Automatic calibration methods, which are objective and relatively easy to be implemented with high speed computers, have become more and more popular in recent years (Vrugt et al., 2003b). Different automatic parameter estimation methods (i.e. PEST, GA, and SCE) have been successfully applied for calibrating SWAT. For example, Govender and Everson (2005) and Wang and Melesse (2005) applied automatic PEST parameter estimation program to estimate hydrologic related variables of SWAT; Muleta and Nicklow (2005) describe using a GA to perform automatic calibration of daily streamflow and sediment yield estimates. SCE is the automatic method that has been incorporated into SWAT2005. Eckhardt and Arnold (2001), Eckhardt et al. (2005), van Griensven and Bauwens (2003; 2005), Vandenberghe et al. (2001), Van Liew et al. (2005; 2007) demonstrated the application of SCE in a broad range of watersheds for streamflow, sediment and pollutant simulation.

For most cases, the SWAT model is calibrated and validated at the drainage outlet of a watershed (Qi and Grunwald, 2005). It was found that spatially distributed calibration and validation accounted for hydrologic patterns in the subwatersheds (Qi and Grunwald, 2005). Using spatially distributed data to calibrate and validate SWAT model is becoming more and more popular (Cao et al., 2006; White and Chaubey, 2005; Vazquez-Amábile and Engel, 2005; and Santhi et al., 2001).

2.4.3 Parameter uncertainty analysis of SWAT

Monte Carlo simulation has been widely used to assess the uncertainty of hydrologic models (Haan and Skaggs, 2003a; 2003b). Several parameter uncertainty analysis

methods have been developed and applied for SWAT. Benaman and Shoemaker (2004) developed a six-step method, which reduced the model output range by an order of magnitude, resulting in reduced uncertainty and the amount of calibration required for SWAT. Muleta and Nicklow (2005) describe a combined method for parameter sensitiviey analysis, calibration, and uncertainty analysis of SWAT. They conducted parameter sensitivity analysis for 35 input parameters, in which LH samples were used to reduce the number of MC simulations needed to conduct the analysis. GA and GLUE methods were then used to conduct parameter calibration and uncertainty analysis of SWAT. Van Greinsven and Meixner (2006) describe several uncertainty analysis tools that have been incorporated into SWAT2005, including a modified SCE algorithm called "parameter solutions" (ParaSol), the Sources of Uncertainty Global Assessment using Split SamplES (SUNGLASSES) which further evaluates results obtained with ParaSol for a different time period (to ascertain bias in the initial confidence region, etc.), and the Confidence ANalysis Of Physical Inputs (CANOPI) which evaluates uncertainty associated with climatic data and other inputs.

2.5 Study area description

In this study, the SWAT model was applied to four watersheds with different climatic and terrain characteristics to test the effectiveness and efficiency of different optimization algorithms. The four watersheds included the YR headwater watershed, MCEW, LREW, and RCEW. The geographic locations of the four watersheds are shown in Figure 2-3 and Figure 2-4. Among the four watersheds, the YR headwater watershed is located in China, and the other three watersheds are located in the United States. These three watersheds are US Department of Agriculture Agricultural Research Service (USDA ARS) experimental watersheds, and have been used in a paper by Van Liew et al. (2007) for testing the suitability of SWAT for the Conservation Effects Assessment Project. The basic characteristics of the four test watershed are described below.

2.5.1 Yellow River headwaters watershed

The Yellow River headwaters watershed (YRHW) is an 114,345 km^2 mountainous river basin, which is located in the northeast part of Tibetan plateau. This area is the

important source of water generation for the Yellow River Basin (Liu, 2004). The average elevation is about 4,217 m, and ranges between 2,600 and 6,266 m. The annual precipitation amount is around 600 mm and the average annual temperature for the YR headwater is near 0ºC. In winter the average temperature is below 0ºC for most of the weather stations, while in summer the average temperature is above 0ºC. This seasonal temperature variation makes snowmelt a significant process in this area (Zhang et al., 2007). This watershed is characterized by gently sloping upland and river bed, and swamp and wetland. The major types of soils in this area are clay and loam with relatively low infiltration rate. The major land cover in the study area is grassland, which accounts for approximately 90% of the total area. Other land use/land cover (forest land, rangeland, agriculture land, and bare area) account for the remaining 10% of the area.

2.5.2 Mahantango Creek Experimental Watershed

Mahantango Creek Experimental Watershed (MCEW) is a tributary of the Susquehanna River in Central Pennsylvania. The MCEW is typical of upland agricultural watersheds within the nonglaciated, folded and faulted, Appalachian Valley and Ridge Physiographic Province (Veith et al., 2005). Climate in the region is temperate and humid, with a long-term average annual precipitation of 1100 mm. The watershed is characterized by shallow, fragipan soils in near-stream areas, and deep, well-drained soils in the uplands (Van Liew et al., 2007). Land use types consist of pasture (38%), forest (34%), mixed croplands (26%), and farmsteads (2%).

2.5.3 Little River Experimental Watershed

The Little River Experimental Watershed (LREW) is the upper 334 km^2 of the Little River and is the subject of long-term hydrologic and water quality research by USDA-ARS and cooperators (Sheridan, 1997). The LREW is located in the Tifton Upland physiographic region, which is characterized by intensive agriculture in relatively small fields in upland areas and riparian forests along stream channels. The region has low topographic relief and is characterized by broad, flat alluvial floodplains, river terraces, and gently sloping uplands (Sheridan, 1997). Climate in this region is characterized as humid subtropical with an average annual precipitation of about 1167 mm based on data

collected by USDA ARS from 1971 to 2000. Soils on the watershed are predominantly sands and sandy loams with high infiltration rates. Since surface soils are underlain by shallow, relatively impermeable subsurface horizons, deep seepage and recharge to regional ground water systems are impeded (Sheridan, 1997). Land use types include forest (65%), cropland (30%), rangeland and pasture (2%), wetland (2%), and miscellaneous (1%).

2.5.4 Reynolds Creek Experimental Watershed

The area of the Reynolds Creek Experimental Watershed (RCEW) is 239 km^2, which is located about 80 km southwest of Boise and exhibits a considerable degree of spatial heterogeneity. The topography of the watershed ranges from a broad, flat alluvial valley to steep, rugged mountain slopes, with a range in elevation from 1101 to 2241 m (Seyfried et al., 2000). Because of orographic effects, the average annual precipitation range from about 250 mm from the outlet to more than 1100 mm at the upper end of the watershed. Perennial streamflow is generated at the highest elevations in the southern part of Reynolds Creek where deep, late lying snowpacks are the source for most water (Seyfried et al, 2000). Although much of the watershed has steep, shallow, rocky soils, there are areas of deep, loamy soils that are rock free. Land cover on Reynolds Creek consists of rangeland and forest communities of sagebrush, greasewood, aspen, and conifers (94%) and irrigated cropland (6%).

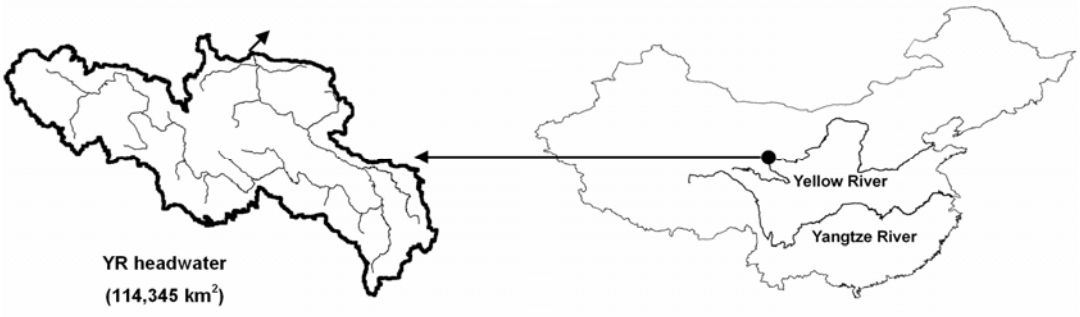

Figure 2-3. Location of the headwaters region of the Yellow River.

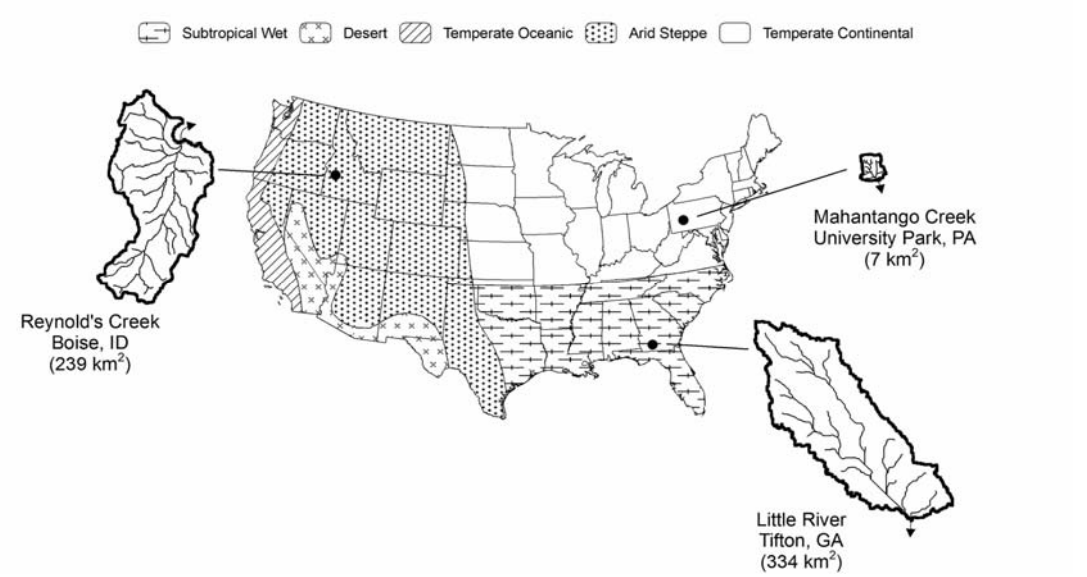

Figure 2-4. Locations of three USDA ARS experimental watersheds. (Modified from Van Liew et al., 2007).

CHAPTER III

SINGLE OBJECTIVE CALIBRATION OF SWAT

3.1 Introduction

Hydrologic models are more and more widely applied by hydrologists and resources managers as a tool to understand and manage natural and human activities that affect watershed systems. The successful application of a hydrologic model depends on how well the model is calibrated (Duan et al., 1992). Hydrologic models, even those physically-based models, often contain parameters that cannot be measured directly due to measurement limits and scale issues (Beven, 2000). These parameters need to be estimated through an inverse method by calibration so that observed and predicted output values are in agreement. Before the widespread availability of high speed computers, hydrologic practitioners utilized knowledge of the watershed and experience with the model to adjust the parameters through a manual trial and error procedure (Gupta et al., 1999). This approach to calibration is subjective and labor intensive. Automatic calibration methods, which are objective and relatively easy to be implemented with high speed computers, have become more and more popular in recent years (Vrugt et al., 2003). Global optimization algorithms, which can efficiently and effectively search optimum parameter solutions that can minimize (or maximize) some objective functions that represent the agreement between observations and model simulations, have been successively applied in the research field of automatic calibration of hydrologic methods. For example, Duan et al. (1992) developed the Shuffled Complex Evolution algorithm (SCE-UA), which has been widely used in hydrologic modeling (Sorooshian et al., 1993) and proved to be consistent and efficient for searching global optimum parameter values of hydrologic models (Vrugt et al., 2003). Other optimization algorithms (i.e., Genetic Algorithms (GA), simulated annealing (SA),

and Levenberg-Marquardt) are also popular methods for automatic calibration of parameters in hydrologic models.

With the popularity of sophisticated physically-based watershed models, the complexity of the calibration problem has increased substantially (Gupta et al., 1998). Although the speed and capacity of computers have increased multi-fold in the past several decades, the time consumed by running hydrologic models (especially those complex, physically based, distributed hydrologic models) is still a concern for hydrologic practitioners. As to which of the available optimization methods that can effectively and efficiently identify good parameter sets is a topic of considerable interest. Several studies have been conducted to evaluate the performance of different algorithms. For example, Cooper et al., (1997) evaluated SCE-UA, GA and SA methods for optimization of the Tank model; Kuczera (1997) compared four search algorithms, SCE-UA, GA, and multiple random start using either simplex or quasi-Newton local searches for parameter optimization of catchment models; Chen et al. (2005) compared the performance of multi-start Powell and SCE-UA methods for calibrating the Tank model; Jha et al., (2006) compared the traditional (Levenberg-Marquardt and Gauss-Newton) and nontraditional (GA) techniques for determining well parameters. The results obtained by the above comparison studies showed that the evolutionary algorithms (SCE-UA and GA) could provide equal or better performance than other methods (Jha et al., 2006; Kuczera, 1997; Chen et al., 1997; Cooper et al., 1997). With the robustness for searching global optimum and ease of implementation, evolutionary algorithms have been widely used in hydrologic modeling. Besides the SCE-UA and GA algorithms, the Particle Swarm Optimization (PSO) has also been used to optimize the arrangement of hydraulic devices in a pipeline system (Jung and Karney, 2006), and train the Artificial Neural Networks' weights for river stage prediction (Chau, 2006). Other evolutionary algorithms, such as Differential Evaluation (DE) (Storn and Price, 1997) and Artificial Immune Systems (AIS) (de Castro and Von Zuben, 2002a; de Castro and Von Zuben, 2002b), although rarely used in hydrologic model calibration, showed promising ability for global optimization of complex systems.

There are many physically-based watershed models that have been successfully applied in practical hydrologic modeling problems. As the time consumed by running these models is enormous huge, it's nearly impossible to test the optimization algorithms for all these complex models. The objective of this paper was therefore to evaluate the efficacy of the five evolutionary algorithms (SCE-UA, GA, PSO, DE, and AIS) for parameter optimization of SWAT. As the time and computational resources did not allow for a vast number of model runs with SWAT, the performance of the five optimization algorithms were only tested for a limited number of evaluations of the model.

3.2 Methods

3.2.1 Global optimization algorithms

The five based global optimization algorithms were investigated in this study. Use of these algorithms depends upon a number of variables, many of which are defined as follows: *D* is defined as the parameter solution dimension; *N* is the number of parameter solutions in the population; $\mathbf{x}_i$ is defined as the *i*th parameter solution in the population, and a *D*-dimensional vector $\mathbf{x}_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$ where x_{id} is the *d*th parameter of the *i*th parameter solution. Different algorithms use different terms to denote parameter solution. The parameter solution is referred to as "chromosome" in GA, "point" in SCE, "particle" in PSO, "antibody" in AIS, and "individual" in DE. The current number of iteration of the algorithms is denoted as *t*. *T* is the maximum number of iterations allowed before optimization is terminated. All five algorithms are population based. A Latin Hypercube algorithm is used to initialize the first population of parameter solutions.

3.2.1.1 GA

GAs are stochastic search procedures inspired by the evolutionary biological processes of natural selection and genetics (Holland, 1975; Goldberg, 1989), such as inheritance, mutation, selection, and crossover. With flexibility and robustness, GAs have been successfully applied to solve complex nonlinear programming problems in many science and engineering branches (Reca and Martinez, 2006), including

hydrologic modeling. For example, Kuo and Liu (2003) applied GAs for optimizing a model for irrigation planning and management; Chang et al. (2005) showed that the GAs provided an adequate, effective and robust way for searching the reservoir operating rule curves. Srivastava (2002) and Arabi et al. (2006) used GAs for optimizing the allocation of watershed management practices. Jia and Culver (2006) optimized total maximum daily load allocations in the Charlottesville Watershed, Virginia using GAs. The general procedures for applying GAs are schematically described in Figure 3-1. There are three major operators in the GAs: selection, crossover, and mutation.

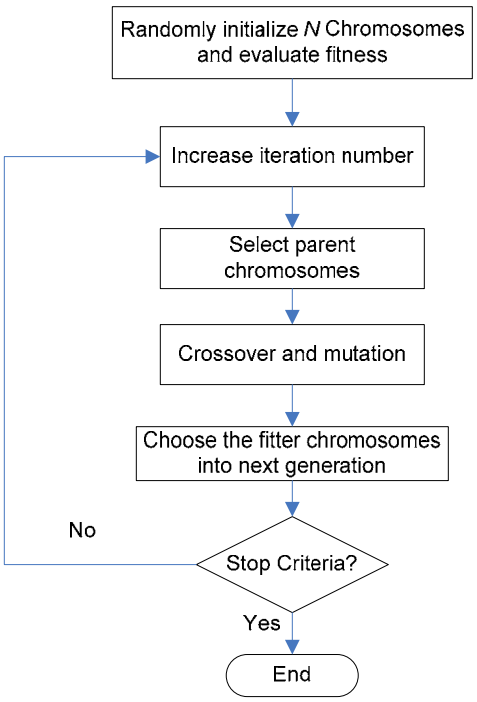

Figure 3-1. Flowchart of the genetic algorithm.

Selection operator. The fitter chromosomes in the population are preferred to be selected to reproduce new promising offspring. A roulette wheel algorithm is applied to select chromosomes for the following crossover and mutation operation. The probability of a chromosome to be selected as parent is proportional to its fitness. In order to

overcome shortcomings of using the original fitness value in the roulette wheel algorithm, a relative fitness value for each chromosome is calculated in the following equation:

$$F'(r) = (f+1)(N+1) - 2rf \quad \textbf{3- 1}$$

Where f is the selection pressure ($0 \le f \le 1$), and r is the rank assigned to the chromosomes based on their fitness.

Crossover operator. The purpose of crossover is exchanging important building blocks of two parent chromosomes to generate new promising offspring. The probability of crossing two chromosomes is determined by an input parameter P_c . There are mainly three crossover methods: 1) one-point crossover, 2) two-point crossover, and 3) uniform crossover (Goldberg, 1989). In this study, the uniform crossover operator has been applied. In the uniform crossover, for each dimension of the parent chromosomes, two parent chromosomes swap the parameter values with a probability of 0.5 to generate a new offspring chromosome.

Mutation operator. For each dimension of the offspring chromosome, a random number between [0,1] is generated. If this random number is less than the mutation probability (P_m), then a newly generated parameter value will replace the old parameter value of the specific dimension.

After the newly bred chromosomes are generated through the selection, crossover, and mutation operators, they will be incorporated into the population using a steady-state-delete-worst plan (Reca and Martinez, 2006), in which the least fit member of the parent population is eliminated and replaced by the offspring. Several control parameters of the GA were determined according to Schaffer et al. (1989) and Reca and Martinez (2006): P_m was set to 1/*D*, P_c was equal to 0.5, and f was set to 1.

3.2.1.2 SCE

The Shuffled Complex Evolution algorithm is very popular for hydrologic model optimization. The SCE algorithm developed by Duan et al. (1992) merges the strengths of the Downhill Simplex procedure (Nelder and Mead, 1965) with the concepts of

controlled random search, systematic evolution of points in the direction of global improvement, competitive evolution (Holland, 1975), and complex shuffling. In a first step of implementation of SCE, an initial population of parameter solutions is randomly sampled for "p" parameters to be optimized. The population is partitioned into several community, each consisting "2p+1" points. Each community is made to evolve based on a statistical "reproduction process" that uses the simplex method, an algorithm that evaluates the objective function in a systematic way with regard to the progress of the search in previous iterations (Nelder and Mead, 1965). At periodic stages in the evolution, the entire population is shuffled and parameter solutions are reassigned to communities to ensure information sharing. As the search progresses, the entire population tends to converge toward the neighborhood of global optimization.SCE searches the entire parameter space and finds the global optimum efficiently and effectively (Sorooshian et al., 1993) and has been successfully used for calibration of SWAT (van Griensven and Bauwens, 2003; Eckhardt et al., 2005; Van Liew et al., 2005; van Griensven and Bauwens 2005). A brief description of the processing sequence of the SCE is presented below (Duan et al., 1992):

1) Initialize N points in the feasible parameter space, and evaluate each parameter solution's fitness.
2) Rank points in ascending order of fitness value.
3) Partition the N points into p complexes, each containing m points. The complexes are partitioned in such a way that the first complex contains every p×*(k-1)+1* ranked point, the second complex contains every $p\times (k-1)+2$ ranked point, and so on, where $k = 1,2,..., m$.
4) Evolve each complex independently by taking β evolution steps:

a) Rank points in each complex in ascending order of fitness value.

b) A sub-complex containing q points is selected according to a prespecified probability distribution using equation (3-2).

$$p_i = \frac{2(m+1-i)}{m(m+1)} \quad i = 1,\ldots,m \qquad \textbf{3- 2}$$

The points with higher fitness value have a higher chance of being chosen to form the sub-complex than the worse points.

c) Rank points in sub-complex in descending order of fitness value, and compute centroid *G* by equation (3-3)

$$G = \frac{1}{q-1}\sum_{j=1}^{q-1} u_j \qquad \textbf{3- 3}$$

u_j is the jth point in current sub-complex.

d) Compute the new point $r = 2G - u_q$ ("reflection").

e) If r is within the feasible parameter space, compute its fitness value f_r and go to procedure (f); otherwise compute the smallest hypercube $H \in R^n$ that contains the points within the current complex, randomly generate a point z within H, compute f_z, and set $r = z$ and $f_r = f_z$ (mutation step).

f) If $f_r > f_q$, replace u_q with r, go to step (l); otherwise compute $c = (G + u_q)/2$ and f_c (contraction step).

g) If $f_c > f_q$, replace u_q with c, go to step (l); otherwise generate a point z within H, compute f_z (mutation step). Replace replace u_q with z.

h) Repeat procedures (c) to (g) α times.

i) Replace the points in sub-complex to current complex using their original positions. Repeat procedures (a) to (h) β times.

5) Shuffle complexes: Combine the points in the evolved complexes into a single sample population; sort the sample population in order of increasing criterion value; re-partition or shuffle the sample population into *p* complexes according to the procedure specified in Step 3. If the number of trials exceeds *T*, stop; else, return to Step 2.

p is number of complexes in a sample population, *m* is number of points in each complex, $p \times m$ equals the population size *N*, *q* is number of points in each sub-complex,

β is number of evolution steps allowed for each complex before complex shuffling, and α is the number of consecutive offspring generated by each sub-complex. According to Duan et al. (1994), m was set to $2\times D + 1$, q was equal to $D + 1$, β was set to $2 \times D + 1$, and α was equal to 1.

3.2.1.3 PSO

Particle swarm optimization (Kennedy and Eberhart, 1995) is a stochastic optimization technique inspired by social behavior of bird flocking or fish schooling. PSO's basic algorithm involves casting a population of particles over the search space. Each particle is assigned to an initially random location and velocity vector, and then each particle adjusts its "flying" according to its own flying experience and its companions' flying experience (Eberhart and Shi, 1998). PSO has been successfully applied to optimize artificial neural networks for river stage prediction (Chau, 2006) and parameter estimation of hydrologic models (Gill et al., 2006b).

Before introducing the PSO algorithm, several variables are defined: $\mathbf{v}_i$ denotes the movement velocity for the ith particle in the swarm, which is a D dimensional vector $\mathbf{v}_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$, v_{id} is the movement velocity of dth parameter of the ith particle; $\mathbf{p}_i$ denotes the personal best position that has been searched by the ith particle, which is also a D-dimensional vector $\mathbf{p}_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$, p_{id} is the value of the dth parameter of the ith particle's personal best position; $\mathbf{p}_g$ denotes the global best position that has been searched by all particles, which is D-dimensional vector $\mathbf{p}_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$. p_{gd} is the value of the dth parameter of the global best particle. Since the first introduction of PSO by Kennedy and Eberhart (1995), several variants of PSO have been proposed. One of the variants that has been successively applied for solving complex optimization problems was applied in this study (Parsopoulos and Vrahatis, 2002). The basic PSO's operation formulas are described as:

$$v_{id}^{\,t+1} = w \cdot v_{id}^{\,t} + \eta_1 \cdot rnd_1 \cdot (p_{id} - x_{id}) + \eta_2 \cdot rnd_2 \cdot (p_{gd} - x_{id}) \qquad \textbf{3- 4}$$

$$x_{id}^{\,t+1} = x_{id}^{\,t} + v_{id}^{\,t+1} \qquad \textbf{3- 5}$$

where η_1 and η_2 are the cognitive and social learning rates, respectively. These two rates control the relative influence of the memory of all the particles and the memory of the individual particle, rnd_1 and rnd_2 are uniformly distributed random numbers between 0 and 1, and w is the inertia weight of previous velocity. Based on previous studies (Shi and Eberhart, 1998; Parsopoulos and Vrahatis, 2002), η_1 and η_2 are both set equal to 2. The initial value of w is set equal to 1.2. The value of w is decreased linearly and set equal to 0.1 for the last iteration. In addition to the above variables, three vectors $\mathbf{x}_{max}$, $\mathbf{x}_{min}$ and $\mathbf{v}_{max}$ are defined to place a limit on the search space and velocities. The general procedures for applying PSO are schematically described in Figure 3-2. For more detailed information of PSO, please refer to Kennedy and Eberhart (2001).

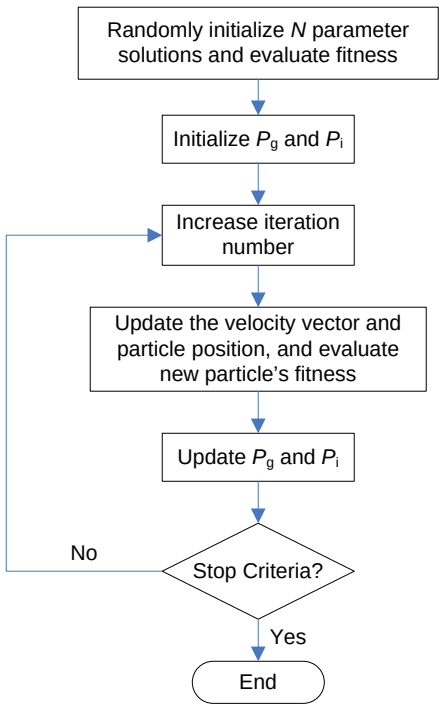

Figure 3-2. Flowchart of the particle swarm pptimization.

3.2.1.4 DE

The DE algorithm is a simple and powerful evolutionary algorithm developed by Storn and Price (1997) for global optimization. The DE has gradually gained popularity and has been applied in many practical cases because of its ease of implementation and good convergence properties. For example, the DE has been successfully applied for multi-sensor fusion (Joshi and Sanderson, 1999), optimization of heat transfer parameters in trickle-bed reactors (Babu and Sastry, 1999), and of surface grinding operations (Krishna, 2007). The DE creates new candidate solutions by combining the parent individual and several other individuals of the same population, and chooses the better one between the new solution and the parent individual into the next generation of the population. The DE algorithm uses three operators to generate new parameter solutions. Mutation and crossover are used to generate new parameter solutions, and the selection operator is used to determine which of the solutions survive into the next generation. There are several variants of DE (Storn and Price, 1997, Krishna, 2007). One variant of DE, noted as DE/rand/1/bin according to Storn and Price (1997), was applied in this study. This variant of DE has been most often used in practice (Brest et al., 2006, Krishna, 2007). The general procedures of DE are illustrated in Figure 3-3.

The mutation, crossover and selection operators are different from those used in GA, and are described as follows:

Mutation operator. For each parameter solution $\mathbf{x}_i$, a mutant parameter vector $\mathbf{v}_i$ is generated according to

$$\mathbf{v}_i = \mathbf{x}_{r1} + F \cdot (\mathbf{x}_{r2} - \mathbf{x}_{r3}) \qquad \textbf{3- 6}$$

where $r1$, $r2$ and $r3$ are random integer numbers from $[1, N]$, and these numbers are different from each other and from the running index i. F is a real number ($F \in [0,2]$) that controls the amplification of the difference vector $(\mathbf{x}_{r2} - \mathbf{x}_{r3})$. If a component of the newly generated parameter vector exceeds the constrained parameter space, then the parameter value will be set to the bound value.

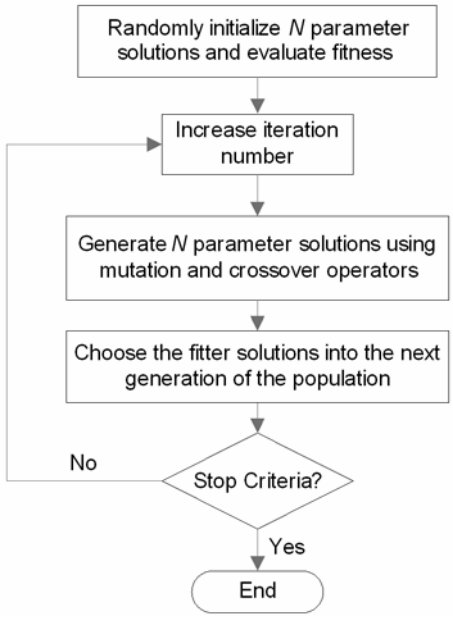

Figure 3-3. Flowchart of the differential evolution.

Crossover operator. The mutated parameter vector $\mathbf{v}_i$ is mixed with the original solution $\mathbf{x}_i$ to produce the trial parameter solution $\mathbf{u}_i = (u_{i1}, u_{i2}, \cdots, u_{iD})$. Each dimension of $\mathbf{u}_i$ is chosen from $\mathbf{v}_i$ or $\mathbf{x}_i$ using following scheme:

$$u_{id} = v_{id} \text{ if } r_d \le CR \text{ or } d = rn(i) \quad \textbf{3-7}$$

$$u_{id} = x_{id} \text{ if } r_d > CR \text{ and } d \ne rn(i) \quad \textbf{3-8}$$

where $r_d \in [0,1]$, $d = 1,2,\ldots D$. CR is the crossover constant which is between [0,1], and $rn(i)$ is a random integer chosen from $\{1,2,\ldots D\}$, which is used to make sure that at least one parameter value of $\mathbf{u}_i$ is from $\mathbf{v}_i$.

Selection operator. The newly yielded $\mathbf{u}_i$ candidate parameter solution is compared with the original solution $\mathbf{x}_i$, and the better one (in terms of fitness value defined by users) will be selected to enter into the next generation of population; otherwise the old solution $\mathbf{x}_i$ will be retained.

For the three control parameters (F, CR, and N) in DE, Storn and Price (1997) suggested that $F \in [0.5,1]$, $CR \in [0.8,1]$, and $N = 10 \cdot D$. In this study, we set the values of F and CR to 0.5 and 0.9 respectively, following the suggestions from Storn and Price (1997), and Brest et al. (2006).

3.2.1.5 AIS

An AIS is a type of optimization algorithm inspired by the principles and processes of the vertebrate immune system. The theory of a biological immune system is beyond the scope of this paper. For further information regarding the theory of an immune system, the reader is referred to de Castro and Von Zuben (2002a) and de Castro and Von Zuben (2002b). The AIS has been successfully applied for optimizing complex systems, like the radial basis function (de Castro and Von Zuben, 2002b), neural networks (Byrski and Kisiel-Dorohinicki, 2005), economic dispatch in power systems (Rahman et al., 2006), and several constrained global optimization problems (Cruz-Cortés et al., 2005). But AIS had seldom been used for optimizing hydrologic models. In this study, the CLONALG (de Castro and Von Zuben, 2002a), a classical AIS algorithm was introduced and applied for parameter optimization of SWAT. In CLONALG, the parameter solution is defined as antibody (**Ab**), and the objective to be optimized is defined as antigen (**Ag**). The flowchart of AIS is shown in Figure 3-4. First of all, a population of N **Ab**s is initiated randomly in the parameter space. Two criteria are used to select promising **Ab**s to reproduce the next generation of candidates. The first one is the **Ab**-**Ag** affinity (expressed as the value of the objective function), and the other one is the **Ab**-**Ab** similarity (expressed as the Euclidean distance). The **Ab** with upper **Ab**-Ag affinity and lower similarity with other Abs are taken as promising solutions. Each **Ab** is assigned to a shared fitness f_s which takes the **Ab**-**Ag** affinity and **Ab**-**Ab** similarity into account:

$$f_s(\mathbf{Ab}_i) = \frac{f(\mathbf{Ab}_i)}{\sum_{j=1}^{N} sh(d(\mathbf{Ab}_i, \mathbf{Ab}_j))} \qquad \textbf{3-9}$$

Where $sh(\cdot)$ is the sharing function, $sh(d) = 1 - (\frac{d}{\sigma_{share}})$, if $d < \sigma_{share}$, otherwise $sh(d) =$ 0. σ_{share} is a threshold value of dissimilarity. The **Ab** s are proliferated through the cloning and hypermutation operator, which perturb the old solution by summing a random vector:

$$\mathbf{Ab}_{i,new} = \mathbf{Ab}_{i,old} + a \cdot \mathbf{x}_{random} \quad a = e^{-\rho \cdot f_s^*} \qquad \textbf{3- 10}$$

where a is the hypermuation rate, $\mathbf{x}_{random}$ is a vector of Gaussian random numbers of mean 0 and standard deviation 1, f_s^* is the normalized value of fitness in the range. The **Ab** s with higher f_s^* are tended to be perturbed with a smaller step size. The total number of new clones generated is determined using the rule suggested by de Castro and Von Zuben (2002a),

$$Nc = \sum_{i=1}^{N} round(\beta \cdot N) \qquad \textbf{3- 11}$$

where Nc is the total number of new **Ab** s, and β is a multiplying factor. For each **Ab**, the number of clones yielded using equation (3-10) is the same. The best n **Ab** s among the combination of newly cloned and the parent **Ab** s with highest fitness are reselected to enter the next generation. Also *N*-*n* new **Ab** s are randomly generated and added to the next generation.

de Castro and Von Zuben (2002a) provides detailed discussion on the application of CLONALG algorithms for parameter optimization. Following the control parameter setting in de Castro and Von Zuben (2002a), σ_{share} was set to 4, n was set to $round(0.9 \cdot N)$, β was 0.1, and ρ was 10.

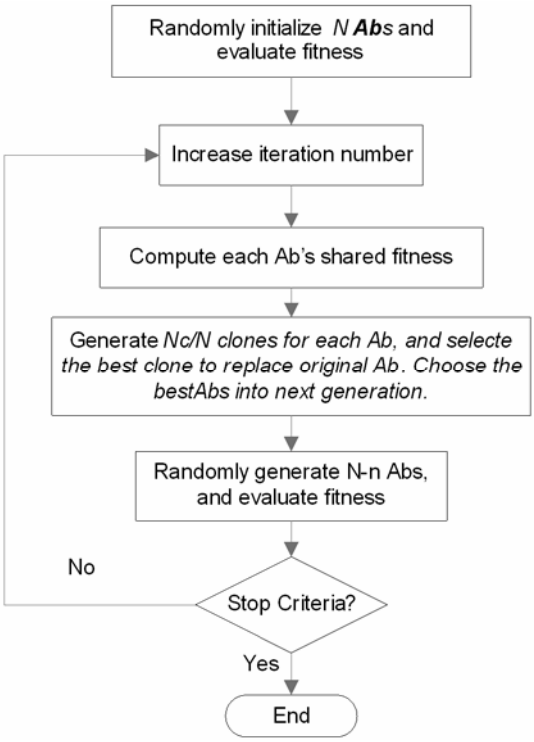

Figure 3-4. Flowchart of the artificial immune system.

3.2.2 Optimization test cases

The performance of the five optimization algorithms is determined by the control parameters. For most of the control parameters, previous literature provides some suggestion on how to choose appropriate settings. Among these control parameters, the population size is an important factor that determines the performance of different algorithms. In this study, most of the control parameters of the five optimization algorithms were set according to recommendations from previous studies. The effect of population size on the performance of different algorithms was further examined with one relatively small population size and one large population size for each optimization algorithm. There are no common criteria for evaluating whether a population size is large or small for different algorithms. The small and large population sizes are different for the five algorithms, and were chosen according to the population sizes that have been tested in previous empirical studies that applied these optimization techniques. The small

population sizes are 66 (two complexes), 50, 30, 10, and 50 for SCE, GA, PSO, AIS and DE, respectively. The large population sizes are 165 (five complexes), 200, 100, 50, and 160 for SCE, GA, PSO, AIS and DE, respectively. There are two optimization cases that were defined for each watershed: 1) small population size scenario and 2) large population size scenario. Hence, there were a total of eight optimization cases for each optimization algorithm that were applied in this study. The definition of the optimization case was denoted using the combination of watershed name and population size, *i.e.*, "watershed name+ population size" was used to represent the optimization cases. For example, "Reynolds + Small" denote that the optimization algorithms were tested on the RCEW with small population size. In this study, the SWAT model was set up for daily flow simulation at the outlets at different watersheds. The calibration periods consists of ten years (1976-1985) in the YR headwater watershed, six years (1995-2000) in MCEW, four years (1995-1998) in LREW, seven years (1966-1972) in RCEW.

3.2.3 Evaluating performance of different algorithms

The optimization objective functions are indicators of agreement between measured and simulate series of the variable of interest. The sum of squares of residuals (SSR) is an often applied objective function in calibrating hydrologic models (Van Liew et al., 2007). In this study, the Nash-Sutcliffe efficiency (E_{ns}), a normalized form of SSR, was selected. The formula to calculate E_{ns} is (Nash and Sutcliffe, 1970; Legates and McCabe, 1999):

$$E_{ns} = 1.0 - \frac{\sum_{i=1}^{N}(O_i - P_i)^2}{\sum_{i=1}^{N}(O_i - \overline{O})^2} \qquad \textbf{3- 12}$$

where P is the model simulated value, O is the observed data, the over bar is the mean for the entire time period of the evaluation, and i = 1, 2, ..., N, where N is the total number of pairs of simulated and observed data. E_{ns} indicates how well the plot of the observed value versus the simulated value fits the 1:1 line, and ranges from $-\infty$ to 1. A new variable, RE_{ns}, defined as the ratio between the average E_{ns} value obtained at

different model evaluation numbers and the best average E_{ns} value obtained after the maximum number of model evaluations was used to represent how fast the algorithm converges. In this study, it's assumed that a RE_{ns} value of 0.99 means the convergence of the algorithm. The five algorithms are all stochastically based, so the results obtained by one trial are stochastic and cannot be used to accurately evaluate the algorithm's performance. The average behavior of multiple trials of each algorithm was used to compare the performance of different algorithms. In this study, ten trials were implemented for each optimization case. Ideally, the optimization algorithm with high average E_{ns} value and small number of model evaluations to reach a E_{ns} value of 0.99 are preferred.

As the time and computer resources are limited, it was not possible to run the computationally intensive model for a very long simulation period or for an unlimited number of model evaluations. The five algorithms were compared based on the average performance of ten trials within a limited and affordable number of model evaluations. On a computer with Pentium IV 3 GHZ and 1GB RAM, the time consumed by one SWAT model evaluation were 30 seconds for the YR headwater watershed, 18 seconds for MCEW, 56 seconds for LREW, 1 minute and 8 seconds for RCEW. According to previous calibration studies of SWAT, usually less than 10000 model evaluations were implemented (van Griensven and Bauwens, 2003; Tolson and Shoemaker, 2007). Considering the time and computer resources availability, the maximum number of model evaluations was limited to 10000 model evaluations for the four test watersheds. The time consumed by one trial were 84 hours in the YR headwater watershed, 50 hours in MCEW, 155 hours and 190 hours for LREW and RCEW, respectively.

Table 3-1. E_{ns} values obtained by different optimization algorithms at different number of model runs in the four test watersheds

		YRHW					MCEW				
		SCE	**GA**	**PSO**	**AIS**	**DE**	**SCE**	**GA**	**PSO**	**AIS**	**DE**
500	Small	0.751	0.782	0.759	0.730	0.758	0.64	0.638	0.691	0.474	0.637
	Large	0.718	0.753	0.768	0.730	0.726	0.628	0.595	0.662	0.576	0.606
1000	Small	0.772	0.799	0.769	0.752	0.783	0.673	0.667	0.694	0.513	0.676
	Large	0.735	0.785	0.780	0.743	0.757	0.656	0.639	0.677	0.577	0.625
2000	Small	0.819	0.815	0.787	0.781	0.803	0.691	0.708	0.7	0.596	0.701
	Large	0.775	0.806	0.801	0.758	0.783	0.687	0.681	0.68	0.588	0.671
3000	Small	0.826	0.822	0.808	0.787	0.810	0.695	0.718	0.7	0.616	0.704
	Large	0.814	0.813	0.811	0.775	0.807	0.69	0.705	0.682	0.612	0.689
4000	Small	0.828	0.824	0.815	0.794	0.819	0.696	0.726	0.7	0.628	0.708
	Large	0.828	0.816	0.814	0.787	0.811	0.692	0.717	0.684	0.619	0.695
5000	Small	0.830	0.826	0.817	0.799	0.825	0.701	0.727	0.7	0.637	0.709
	Large	0.830	0.820	0.816	0.791	0.818	0.693	0.719	0.689	0.633	0.698
10000	Small	0.833	0.829	0.819	0.803	0.829	0.704	0.735	0.7	0.671	0.713
	Large	0.831	0.825	0.827	0.806	0.830	0.696	0.733	0.703	0.659	0.707
		LREW					**RCEW**				
		SCE	**GA**	**PSO**	**AIS**	**DE**	**SCE**	**GA**	**PSO**	**AIS**	**DE**
500	Small	0.731	0.685	0.694	0.487	0.715	0.64	0.638	0.691	0.474	0.637
	Large	0.704	0.673	0.725	0.569	0.645	0.628	0.595	0.662	0.576	0.606
1000	Small	0.757	0.717	0.713	0.503	0.759	0.673	0.667	0.694	0.513	0.676
	Large	0.723	0.707	0.752	0.583	0.685	0.656	0.639	0.677	0.577	0.625
2000	Small	0.774	0.747	0.774	0.565	0.783	0.691	0.708	0.7	0.596	0.701
	Large	0.775	0.739	0.771	0.626	0.733	0.687	0.681	0.68	0.588	0.671
3000	Small	0.774	0.766	0.783	0.609	0.794	0.695	0.718	0.7	0.616	0.704
	Large	0.782	0.757	0.776	0.634	0.768	0.69	0.705	0.682	0.612	0.689
4000	Small	0.774	0.775	0.783	0.632	0.795	0.696	0.726	0.7	0.628	0.708
	Large	0.783	0.764	0.778	0.637	0.782	0.692	0.717	0.684	0.619	0.695
5000	Small	0.775	0.779	0.784	0.662	0.797	0.701	0.727	0.7	0.637	0.709
	Large	0.783	0.771	0.779	0.639	0.793	0.693	0.719	0.689	0.633	0.698
10000	Small	0.78	0.782	0.784	0.707	0.802	0.705	0.735	0.7	0.671	0.713
	Large	0.784	0.784	0.781	0.65	0.802	0.694	0.733	0.703	0.659	0.707

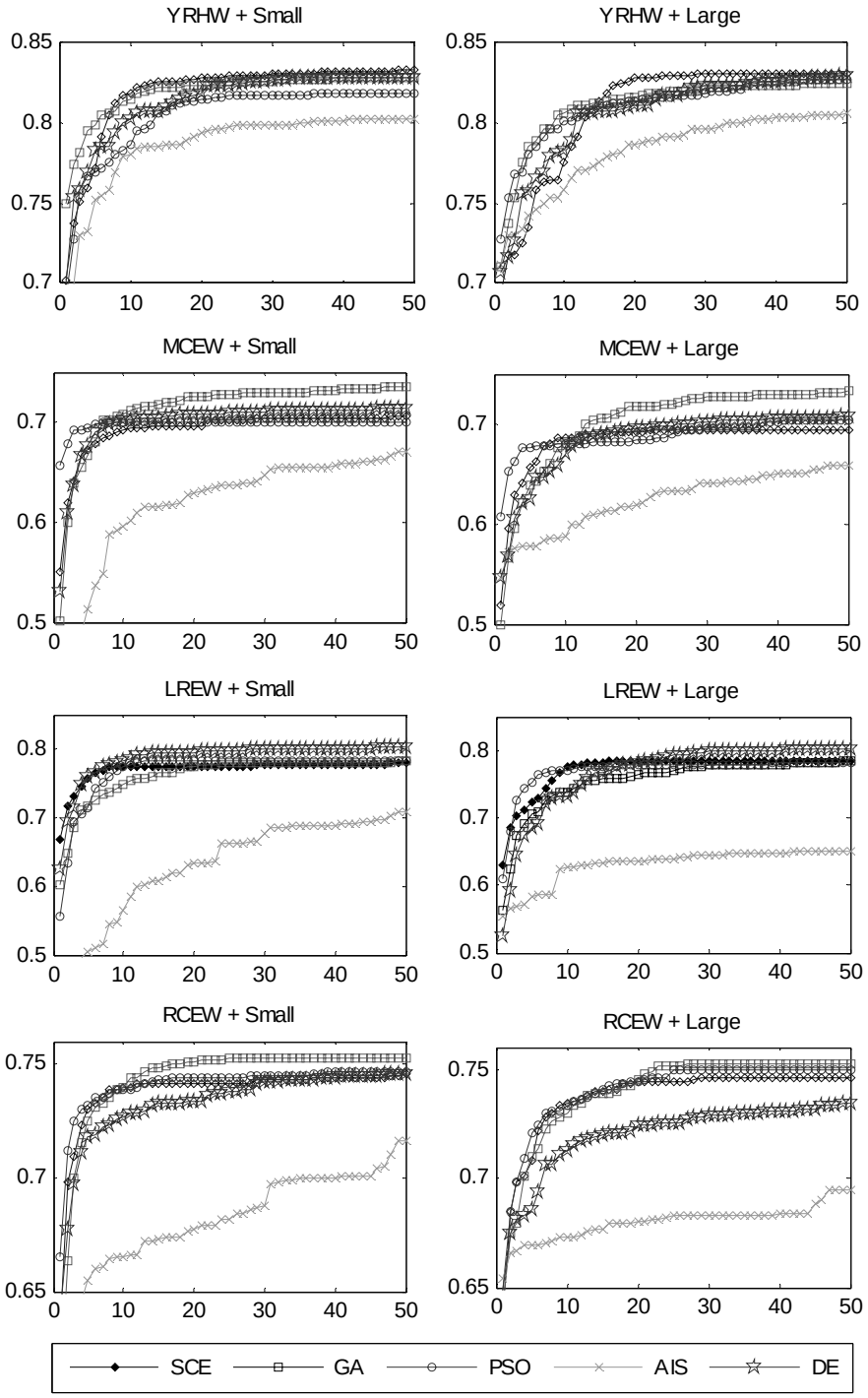

Figure 3-5. Performance of different optimization algorithms versus evaluation number in the four test watersheds.

3.3 Results and discussion

The curves of average objective function values against model evaluation number obtained by different algorithms with large and small population size are shown in Figure 3-5 for YR headwaters watershed, MCEW, LREW and RCEW, respectively. In Figure 3-5, the x axis represents the model evaluation number × 200, y axis represents E_{ns} value. The average objective values and relative performance ranks of different algorithms at different model evaluation numbers are listed in Table 3-1 for YRHW, MCEW, LREW and RCEW, respectively. Based on these figures and tables, the performances of different algorithms in the four test watersheds were analyzed and presented in the following sections. For most cases, AIS performed the least among the five algorithms. The analysis was mainly focused on SCE, GA, PSO and DE.

3.3.1 Performances of different algorithms in YRHW

The selected optimization algorithms exhibited various performance levels at different model evaluation numbers (Figure 3-5 and Table 3-1). From Table 3-1, after 10000 model runs, the best average E_{ns} values obtained by the selected algorithms are 0.833 (SCE with large population), 0.829 (GA with small population), 0.827 (PSO with large population), and 0.830 (DE with small population). Although the four algorithms obtained close objective values with large number of model runs, they exhibited very different performance levels at small number of model evaluations. One algorithm may be preferred for small number of model evaluations while another algorithm may be preferred for large number of model runs. For example, GA found better objective values with small number of model runs (500 and 1000), while SCE obtained better results given large number of model evaluations (more than 2000). The differences between the best average E_{ns} values obtained by different algorithms at small number of model runs are larger than those obtained with larger number model evaluations. For example, the maximum difference between the best final average E_{ns} values obtained by SCE, GA, PSO and DE was 0.006, while this difference was 0.031 given 500 model runs.

It was also found that, the objective values change relatively quickly for the initial 1000 model evaluations, and begin to change relatively slow after that. The RE_{ns} values

at small number of model evaluations represent the capacity of each algorithm to approach to objective values that can be obtained by each scheme with 10000 model evaluations within limited computational time. In the YR headwater watershed, all the algorithms reached RE_{ns} values larger than 0.86 and 0.88 for 500 and 1000 model runs, respectively. Based on the RE_{ns} values obtained with limited model runs, the objective values obtained by each algorithm with large number of model evaluations can be roughly estimated. In general, each scheme needs less than 5000 model evaluations to reach a RE_{ns} value of 0.99 and approximate the best objective value that can be obtained by each algorithm with 10000 model evaluations.

The effect of population size on average E_{ns} values obtained by the optimization algorithms was relatively stronger for the initial 5000 model evaluations than for the model runs after 5000. For example, the difference between average objective values obtained by SCE with small and large population sizes was 0.044 at 2000 model evaluations, while these differences were within 0.012 for all optimization algorithms after 5000 model evaluations.

3.3.2 Performances of different algorithms in MCEW

Figure 3-5 shows that the performance of the GA is much better than the other algorithms after the initial 3000 model evaluation. The GA exhibited an average E_{ns} value larger than 0.72 with 5000 model evaluations, while the other algorithms did not reach E_{ns} values of 0.72 even after 10000 model evaluations. Within 1000 model runs, PSO performed much better than other algorithms (Table 3-1). The maximum difference between the final average E_{ns} values obtained by SCE, GA, PSO and DE was 0.032 after 10000 model evaluations. This shows that different optimization algorithms can obtain substantially different objective values even after large number of model runs. In the MCEW, RE_{ns} values reached 0.99 for all optimization algorithms within 5500 model evaluations, except for the AIS (Table 3-1). With 500 model runs, SCE and PSO obtained RE_{ns} values larger than 0.9, and GA and DE obtained RE_{ns} values larger than 0.8. For 1000 model runs, SCE and PSO obtained RE_{ns} values larger than 0.95, and GA and DE obtained RE_{ns} values larger than 0.85.

For small number of model runs, the difference between the objective values obtained by the each algorithm with large or small population size was relatively larger than that for large number of model runs. For instance, the differences between the final average objective values obtained by SCE, GA, PSO and DE with small and large population sizes were within 0.007, while this difference obtained by GA with small and large population sizes reached 0.043 at 500 model runs.

3.3.3 Performances of different algorithms in LREW

With 10000 model evaluations, the final average E_{ns} values obtained by the selected algorithms were 0.784 (SCE with large population), 0.784 (GA with small population), 0.784 (PSO with small population), and 0.802 (DE with small population), respectively (Table 3-1). SCE obtained better objective values than other algorithms with 500 model runs, and DE obtained better objective values after 1000 model runs. To reach RE_{ns} values of 0.99, SCE and PSO need 2000 model evaluations, and GA and DE need 5000 model evaluations. With 500 model runs, SCE and PSO obtained RE_{ns} values larger than 0.89, and GA and DE obtained RE_{ns} values larger than 0.8. For 1000 model runs, SCE, GA, and PSO obtained RE_{ns} values larger than 0.9, and DE obtained RE_{ns} values larger than 0.85.

The differences between final average E_{ns} values obtained by different algorithms with small or large population size were within 0.004, which show that the E_{ns} values are not sensitive to population size after large number of model runs in the LREW. But this difference was 0.06 for DE with small and large populations at 500 model runs.

3.3.4 Performances of different algorithms in RCEW

After 10000 model evaluations, the GA with small population size exhibited the best objective function value (0.753), followed by PSO (0.751), SCE with large population size (0.746), and DE with small population size (0.746), respectively (Table 3-1). For the initial 500 or 1000 model runs, PSO obtained better results than other algorithms. The maximum difference between the best final average E_{ns} values obtained by SCE, GA, PSO and DE was 0.007, and this difference is 0.027 at 500 model runs. In the RCEW, to reach RE_{ns} values of 0.99, 2600 model evaluations were required for the SCE and PSO,

and 4500 for the GA and DE. All the algorithms reached RE_{ns} values larger than 0.90 for 500 model runs, and 0.93 for 1000 model evaluations.

Similar to previous results obtained in the above three test watersheds, the differences between final average E_{ns} values obtained by different algorithms with small or large population size were relatively small (within 0.008), and these differences were relatively large for the initial model runs.

Table 3-2. performance ranks of different optimization algorithms at different number of model evaluations in the four test Watersheds

		SCE	GA	PSO	DE			SCE	GA	PSO	DE
YRHW	500	4	1	2	3	LREW	500	1	4	2	3
	1000	4	1	3	2		1000	2	3	4	1
	2000	1	2	4	3		2000	2	4	3	1
	3000	1	2	3	4		3000	3	4	2	1
	4000	1	2	4	3		4000	2	4	2	1
	5000	1	2	4	3		5000	3	4	2	1
	10000	1	3	4	2		10000	2	2	2	1
MCEW	500	2	3	1	4	RCEW	500	2	3	1	4
	1000	3	4	1	2		1000	2	3	1	4
	2000	4	1	3	2		2000	1	1	3	4
	3000	4	1	3	2		3000	3	1	2	4
	4000	4	1	3	2		4000	2	1	2	4
	5000	3	1	4	2		5000	3	1	2	4
	10000	3	1	4	2		10000	3	1	2	4

3.4 Discussion

The results obtained in previous sections show that no one optimization algorithm can consistently perform better than the other algorithms for the selected test watersheds. To some extent, this indicates the complexity and difficulty of parameter optimization for SWAT model. Although all the test cases used SWAT as the model for parameter calibration, it appears as though the properties of the four optimization cases are

different from each other and this leads to evidently different performances of the selected algorithms. The overall performances of the five optimization algorithms, and the influence of model evaluation number and population size, were discussed in the following sections.

Using the best final average E_{ns} values obtained by the each of the selected algorithms as the indicator of performance, the performance ranks of the algorithms in the four test watersheds are shown in Table 3-2. The GA performed best for the RCEW and MCEW, DE performed best for LREW, and SCE performed best for YRHW. Using the cumulative rank as the indicator of the comprehensive performance in the four test watersheds (Figure 3-6), GA performed the best in terms of finding good objective values with large number of model runs, followed by DE, SCE, PSO and AIS.

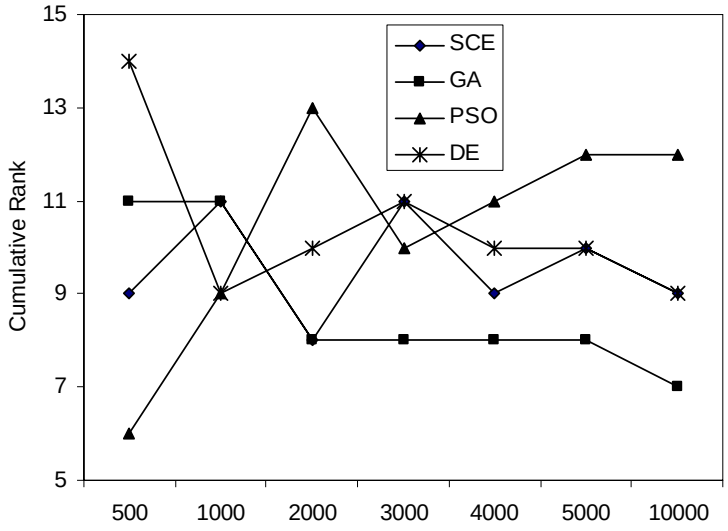

Figure 3-6. Cumulative performance ranks of four optimization algorithms at different number of model evaluations in the four test watersheds

For computationally intensive models, the number of model evaluations needed for obtaining acceptable objective values is an important factor for selecting the optimization algorithm. The SWAT model of detailed characterization of large river basin can take hours or days to implement once. For this type of cases, it is difficult to run the model for large number. The algorithms that can find better objective values within limited model evaluations (less than 1000) are preferred. The performance ranks of different algorithms evaluated with the best average E_{ns} values obtained by each algorithm at different number of model evaluations are listed in Table 3-2. As the AIS can not get good results comparable to the other optimization algorithms, it was not discussed here. It is apparent that the performance of the selected optimization algorithms could change appreciably with model evaluation numbers and watershed characteristics. The cumulative performance ranks of the four optimization algorithms at different number of model evaluations in the four test watersheds are shown in Figure 3-6. It is revealed that PSO performed best with 500 model runs, PSO and DE performed best with 1000 model runs, SCE and GA performed best with 2000 model runs, and GA performed best with more than 2000 model runs. PSO is the preferred choice for less than 1000 model evaluations. For most optimization cases, PSO can obtain RE_{ns} values larger than 90% with 500 or 1000 model runs, which can be taken as fairly good approximation of the best values obtained by PSO after 10000 model runs. In general, results show that SCE and PSO converge faster than GA and DE. The numbers of model evaluations required by various optimization algorithms to obtain a RE_{ns} value of 0.99 are summarized with a conservative consideration of the convergence numbers of the four optimization techniques with small and large population sizes in the four test watersheds. Overall, SCE, GA, PSO and DE need no more than 3200, 5400, 4400, and 4800 model runs, respectively.

It is worthy noting that the population size could exert influence on the performance of the various algorithms. As the difference between the E_{ns} values obtained by each optimization technique due to using small or large population size at small number of model evaluations is less than that at small number of model evaluations, the selection of population size is mainly based on the performances of different algorithms at fewer number of model evaluations (less than 3000). In general, small population size provided better objective function values than large population size for SCE, GA, DE, and PSO for fewer model evaluation (Table 3-1). In the future application of these algorithms for optimizing SWAT, small population size is proffered.

The results discussed above, to some extent, agree with the popular no free lunch (NFL) theorem that "for any optimization algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class" (Wolpert and Macready, 1997). In general, GA performed better than the other algorithms in terms of finding good average E_{ns} values, on the other hand, PSO need less model runs to find acceptable objective values than other algorithms. Although AIS performed the least in terms of both finding best E_{ns} values and efficient convergence to good objective values, it can search multiple local optimum simultaneously, which could be another attracting property for hydrologic model calibration and deserve further analysis in the future. Similar results were also obtained by other numerical evaluation of different global optimization algorithms. For instance, based on the comparison of five stochastic global optimization algorithms, Ali et al. (2005) concluded "one algorithm may be preferred if a small number of function evaluations is allowed but a different algorithm may be favored if a large number of function evaluations is permitted". Although GA algorithm exhibits the better comprehensive rank in terms of finding good average E_{ns} values, it is not possible to infer that this algorithm will always provide the better performance on parameter calibration of SWAT model. The GA can be taken as the first choice when the modelers are interested in finding global optimum, while when the modelers are interested in obtaining acceptable good calibration results within limited computation budget the PSO may be better choice.

3.5 Summary

Efficient and effective algorithms for optimization of computationally intensive hydrologic models like SWAT are becoming increasingly more important because of limited time and computational resources. The purpose of this study was to evaluate the performance of five optimization algorithms for parameter calibration of SWAT within the context of limited model evaluations. In this study, five global optimization algorithms (SCE, GA, PSO, AIS and DE) were tested for parameter calibration of SWAT in four watersheds. For future application of SWAT across United States and other watersheds worldwide, several empirical recommendations on selecting optimization algorithms for SWAT are provided based on the overall performances of different optimization algorithms in the four test watersheds. The GA outperform the other four algorithms given model evaluation numbers larger than 2000, while PSO can obtain better parameter solutions than other algorithms given fewer number of model runs (less than 2000). Given limited computational time, the PSO algorithm is preferred, while GA should be chosen given plenty of computational resources. If GA is chosen to optimize SWAT with large number of model evaluations, the performances of GA can not be pronouncedly improved after 5400 model runs. When applying PSO and GA to calibrate parameters of SWAT, small population size is preferred. It is also worth noting that different optimization algorithms exhibited various preferred properties, incorporating the strength of different algorithms into one powerful algorithm seems to hold promise for future investigations.

CHAPTER IV

MULTI-SITE CALIBRATION OF SWAT

4.1 Introduction

In recent years, hydrologic models have been increasingly used by hydrologists and water resources managers to understand and manage natural and human activities that affect watershed systems. These hydrologic models can contain parameters that cannot be measured directly due to measurement limitations and scaling issues (Beven, 2000). For practical applications in solving water resources problems, model parameters are calibrated to produce model predictions that are as close as possible to observed values. When calibrating a hydrologic model, one or more objectives are often used to measure the agreement between observed and simulated values. The objectives to be optimized can be the combination of multiple goodness-of-fit estimators (e.g. relative error, coefficient of determination), multiple variables (e.g. water, energy, sediment, and nutrients), and multiple sites (Yapo et al., 1998; Gupta et al., 1998; Santhi et al., 2001b; Van Liew and Garbrecht, 2003; White and Chaubey, 2005; Demarty et al., 2005; Cao et al., 2006; Engeland et al., 2006; Bekele and Nicklow, 2007). With the recent development of distributed hydrologic models which can spatially simulate hydrologic variables, the use of multi-site observed data to evaluate model performance is becoming more common.

In the application of SWAT, multi-site data have been used to calibrate parameter values (Santhi et al., 2001b; Van Liew and Garbrecht, 2003; White and Chaubey, 2005; Cao et al., 2006; Bekele and Nicklow, 2007). For simultaneous multi-site automatic calibration of SWAT, two types of calibration methods are usually implemented. The first calibration method aggregates the different objective function values calculated at each monitoring site into one integrated value, and then to apply the single objective

optimization algorithms for parameter estimation (e.g., van Griensven and Bauwens, 2003). The second calibration method uses multi-objective evolutionary algorithms to optimize the different objective functions calculated at multiple sites simultaneously, and finds a set of multiple Pareto optimal solutions (e.g., Bekele and Nicklow, 2007). Currently, the Shuffled Complex Evolution (SCE) algorithm (Duan et al., 1992) is incorporated into SWAT for automatic parameter estimation using a single objective (one objective function or integrated multiple objective functions) (van Griensven and Bauwens, 2003). In many SWAT applications, the model has been calibrated using objective functions at single site or for integrated multi-site objective functions, but multi-objective evolutionary algorithms were seldom applied for multi-site calibration. Therefore, the objective of this study was to compare and evaluate the effect of single and multi-objective optimization schemes on the calibrated parameter values and simulated hydrographs from SWAT. In order to accomplish this objective, a program for parameter optimization of SWAT using single and multi-objective evolutionary algorithms was developed. The single objective and multi-objective optimization algorithms applied in this study were a Genetic Algorithms (GA) and a Strength Pareto Evolutionary Algorithm 2 (SPEA2), respectively. These two optimization algorithms were implemented to estimate the parameters in SWAT for the Reynolds Creek Experimental Watershed in Idaho with observed streamflow data at three monitoring sites. The differences between estimated parameter values and simulated hydrographs are explored and discussed. The results of this study are expected to help the users of SWAT and other distributed hydrologic models understand the sensitivity of distributed hydrologic simulation to different calibration methods and to show the advantages and disadvantages of single objective and multi-objective parameter estimation methods.

4.2 Material and methods

4.2.1 Study area description

The RCEW was selected as the case study area of this multi-site calibration of SWAT. The locations of the RCEW and three streamflow monitoring gages (Salmon, Tolgate, and Outlet) are shown in Figure 4-1. For modeling purposes, the RCEW was

partitioned into subwatersheds connected by a stream network and then into HRUs consisting of unique combinations of land cover and soils in each subwatershed.

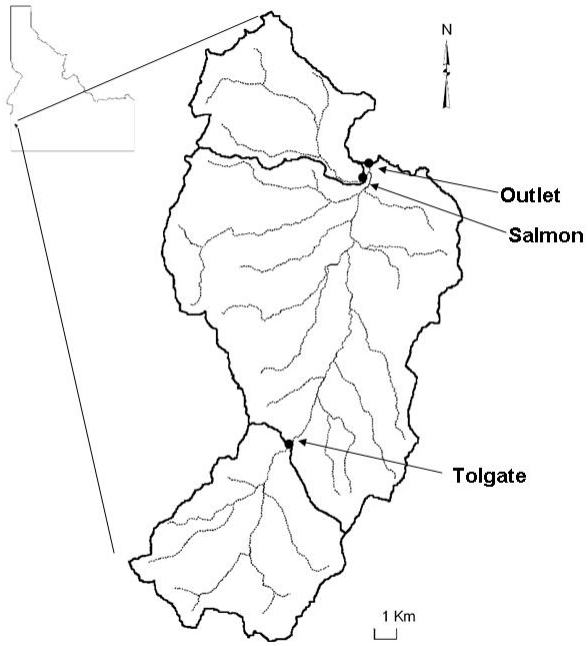

Figure 4-1. Location of RCEW and three streamflow monitoring gages.

4.2.2 Optimization algorithms

4.2.2.1 Single objective optimization

For single objective optimization, there is only one objective function that needs to be optimized. Given the objective function $f:\Omega \subseteq R^D, \Omega \neq \Theta$, for $\mathbf{x} \in \Omega$ the value $f^* = f(\mathbf{x}^*) < \infty$ is called a global maximum if and only if $\forall \mathbf{x} \in \Omega : f(\mathbf{x}^*) \geq f(\mathbf{x})$, where, $\mathbf{x}^*$ is the parameter solution for global maximum and the set Ω is the feasible parameter space.

There are many automatic calibration algorithms that can be used to implement the single objective optimization. Based on the comparison of SCE, GA, PSO, DE and AIS

in Chapter III, and GA is selected as the single objective optimization algorithm for SWAT.

4.2.2.2 Multi-objective optimization

Before the introduction of multi-objective optimization algorithms, several basic concepts related to multiple objective optimization are introduced below (Coello Coello et al., 2004):

Definition 1 [General Multi-objective Optimization Problem (MOP)]: Find the vector $\mathbf{x}^* = [x_1^*, \ldots x_n^*]$ which will satisfy the l inequality constraints

$$g_i(\mathbf{x}) \geq 0 \quad i = 1,2,\ldots l \qquad \textbf{4-1}$$

the p equality constraints

$$h_i(\mathbf{x}) = 0 \quad i = 1,2,\ldots p \qquad \textbf{4-2}$$

And will optimize the vector function

$$\vec{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_x(\mathbf{x}), \ldots, f_m(\mathbf{x})] \qquad \textbf{4-3}$$

Definition 2 (Pareto optimality): A point $\vec{x}^* \in \Omega$ is Pareto optimal if for every $\mathbf{x} \in \Omega$ and $I = \{1,2,\ldots m\}$ either

$$\forall_{i \in I}(f_i(\mathbf{x}) = f_i(\mathbf{x}^*)) \qquad \textbf{4-4}$$

Or, there is at least one $i \in I$ such that

$$f_i(\mathbf{x}) > f_i(\mathbf{x}^*) \qquad \textbf{4-5}$$

Definition3 (Pareto Dominance): A vector $\mathbf{u} = (u_1, u_2, \ldots u_m)$ is said to dominate $\mathbf{v} = (v_1, v_2, \ldots v_m)$ (denoted by $\mathbf{u} \prec \mathbf{v}$) if and only if $\mathbf{u}$ is partially less than $\mathbf{v}$, i.e., $\forall i \in \{1,\ldots m\}, u_i \leq v_i \wedge \exists i \in \{1,\ldots m\} : u_i < v_i$.

Definition 4 (Pareto Optimal Set): for a given MOP $\mathbf{f}(\mathbf{x})$. The Pareto optimal set (P^*) is defined as:

$$P^* := \{\mathbf{x} \in \Omega \mid \neg \exists \mathbf{x}' \in \Omega, \mathbf{f}(\mathbf{x}') \prec \mathbf{f}(\mathbf{x})\} \qquad \textbf{4-6}$$

Definition 5 (Pareto front): fro a given MOP $\vec{f}(\vec{x})$ and Pareto optimal set P^*, the Pareto front (PF^*) is defined as

$$PF^* := \{\mathbf{u} = \mathbf{f} = (f_1(\mathbf{x}),\ldots,f_m(\mathbf{x})) \mid \mathbf{x} \in P^*\} \quad \textbf{4- 7}$$

For multi-objective optimization problems (MOP), a series of objective functions need to be taken into account simultaneously. The general multi-objective optimization problem can be defined as: find the parameter solution $\mathbf{x}^*$ that will optimize the objective function vector $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_x(\mathbf{x}),\ldots,f_m(\mathbf{x})]$ where m the number of objective functions is. As there are multiple objective functions that need to be optimized simultaneously, and different objective functions prefer different parameter solutions, it is difficult to find a single global optimum parameter solution. The Pareto optimality concept is defined to evaluate whether a parameter set is "optimal" or not. For a objective function vector $\mathbf{f}(\mathbf{x}') = [f_1(\mathbf{x}'), f_2(\mathbf{x}'),\ldots,f_m(\mathbf{x}')]$, it is said to dominate another objective function vector $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}),\ldots,f_m(\mathbf{x})]$ (denoted by $\mathbf{f}(\mathbf{x}') \succ \mathbf{f}(\mathbf{x})$), if $\forall i \in \{1,\ldots m\}, f_i(\mathbf{x}') \geq f_i(\mathbf{x}) \wedge \exists i \in \{1,\ldots m\} : f_i(\mathbf{x}') > f_i(\mathbf{x})$. If the objective function vector $\mathbf{f}(\mathbf{x}^*)$ of a point $\mathbf{x}^* \in \Omega$ is not dominated by all the other objective function vectors of the parameter solutions in the feasible parameter space, then $\mathbf{x}^*$ is taken as a Pareto optimal parameter solution. An illustration of the PF^* and dominated objective function vectors is shown in Figure 4-2, where sold circle denotes dominated objective function vectors, and empty circles consist of the PF*.

The purpose of multi-objective optimization is to search the feasible parameter space and find those parameter solutions which are Pareto optimal.

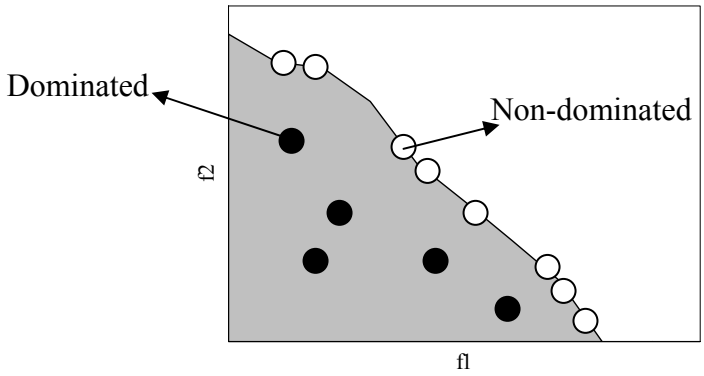

Figure 4-2. Illustration of the PF* and dominated objective function vectors. (From Zitzler et al., 2002).

Among many multi-objective optimization algorithms that have been presented and successfully applied, the SPEA2 (Zitzler et al., 2002) was adopted here to conduct parameter estimation in SWAT model according to the comparison of several state-of-the-art MOAs (Tang et al., 2006). In order to give a clear description of the SPEA2, several new symbols are defined: $\overline{N}$ is the external archive size, and A is the Pareto optimal set. Different from the single objective GA, the SPEA2 applies different procedures to calculate the fitness of each parameter solution and keep the diversity of the candidate parameters. The basic procedures of implementing the SPEA2 are illustrated in Figure 4-3 and the following sections (Zitzler et al., 2002):

Initialization. Generate an initial population P_0 and create the empty archive (external repository) $\overline{P}_0 = \Theta$. Set $t = 0$.

Fitness assignment. In order to calculate fitness values of chromosomes in P_t and $\overline{P}_t$, three major steps need to be implemented. First of all, each chromosome i in the archive $\overline{P}_t$ and the population P_t is assigned a strength value $S(i)$ representing the number of solutions it dominates. Second, on the basis of S values, the raw fitness $R(i)$ of a chromosome i is calculated as $R(i) = \sum_{j \in P_t + \overline{P}_t, j \succ i} S(j)$. $R(i) = 0$ represents a non-

dominated chromosome, while a high $R(i)$ value means that i is dominated by many chromosomes. Third, the raw fitness $R(i)$ needs to be adjusted to incorporate the density of each chromosome, in case many chromosomes have the same raw fitness when most chromosomes do not dominate each other. The density of each chromosome is calculated using the k-th nearest neighbor method, which defines the density of a chromosome as a function of its distance to the k-th nearest neighbors in the objective space (σ_i^k) (Zitzler et al., 2002). In SPEA2, k is set equal to $\sqrt{N+\overline{N}}$. Then the density of each chromosome i is defined as $D(i)=\frac{1}{\sigma_i^k+2}$, where the number "2" is added to the denominator to ensure that $D(i)$ is less tan 1 (Zitzler et al., 2002). Finally, adding $D(i)$ to the raw fitness value $R(i)$ yields each chromosome's fitness $F(i)=R(i)+D(i)$.

Environmental selection. Copy all Pareto optimal chromosomes in P_t and $\overline{P}_t$ to $\overline{P}_{t+1}$. If the size of $\overline{P}_{t+1}$ exceeds $\overline{N}$, then reduce $\overline{P}_{t+1}$ by means of truncating the non-dominated chromosomes with less fitness $F(i)$, otherwise if the size of $\overline{P}_{t+1}$ is less than $\overline{N}$, then fill $\overline{P}_{t+1}$ with best dominated chromosomes in P_t and $\overline{P}_t$.

Termination. If $t > T$ or another stopping criterion is satisfied then set A to the set of parameter vectors represented by the non-dominated chromosomes in $\overline{P}_{t+1}$. Stop.

Mating selection and variation. Perform tournament selection with replacement on $\overline{P}_{t+1}$ to fill the mating pool. Apply crossover and mutation operators to the mating pool and set P_{t+1} to the resulting population. Increment generation counter ($t=t+1$) and go to the fitness assignment step.

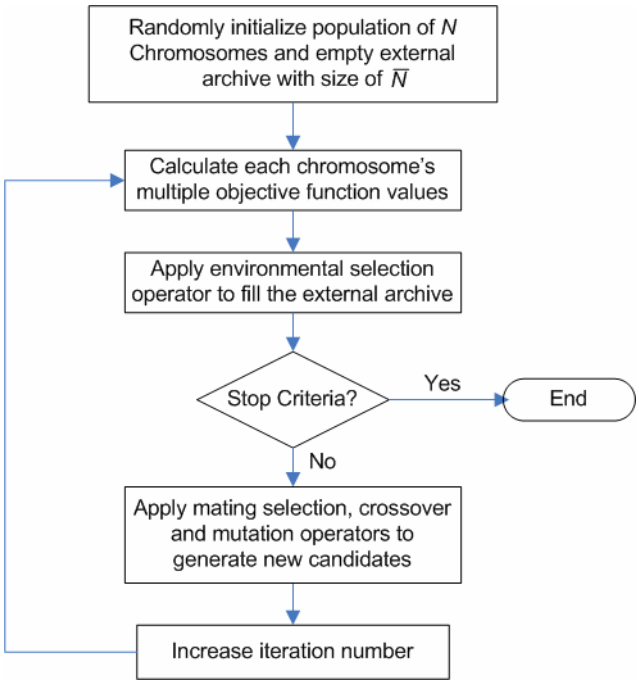

Figure 4-3. Flowchart of the SPEA2.

4.2.3 Experimental test design

In order to understand the effect of applying different parameter optimization schemes (single objective and multi-objective optimization techniques) on the calibrated parameter values and simulated hydrographs of the SWAT model, several GA based optimization schemes were designed for comparative purposes, in addition to the application of the multi-objective optimization scheme (SPEA2). The GA with the optimization objective function of E_{ns} at Salmon is denoted as GA-sal. Similarly, GA-tol denotes GA with objective function of E_{ns} at Tolgate, GA-out denotes GA with objective function of E_{ns} at Outlet, and GA-sum denotes GA with the summed objective functions at all three monitoring stations. There are a total of five optimization cases, which include the GA-sal, GA-tol, GA-out, GA-sum and SPEA2. As SPEA2 can optimize all three objective functions at Salmon, Tolgate and Outlet simultaneously and find a set of Pareto optimal solutions, several representative Pareto optimal solutions were selected for analysis and comparison: SPEA2-sal, SPEA2-tol, SPEA2-out, and

SPEA2-sum denote the parameter solutions with best objective function values at Salmon, Tolgate, Outlet and the sum of all three monitoring stations, respectively. The comparison between the different optimization schemes was mainly based on the eight representative parameter sets (GA-sal, GA-tol, GA-out, GA-sum, SPEA2-sal, SPEA2-tol, SPEA2-out, and SPEA2-sum). The GA and SPEA2 algorithms are stochastically based, so the results obtained by one trial are stochastic. Based on previous studies of applying evolutionary optimization algorithms for SWAT, all the optimization schemes were run as 10 trials (10000 model evaluations for each trial) to obtain the optimized parameter solutions and objective function values.

4.3 Results and discussion

4.3.1 Optimized objective function values, parameter sets and hydrographs by different optimization schemes

The objective function values at Salmon, Tolgate and Outlet obtained by the GA-sal, GA-tol, GA-out, GA-sum, and SPEA2 schemes are shown in Figure 4-4. For each single objective optimization scheme, a best objective function value was obtained, while 96 Pareto optimal objective function vectors were obtained for the multi-objective optimization scheme. For each objective function, the values obtained by the different optimization schemes are listed in Table 4-1. Simulations show that the single objective optimization schemes can identify better values for each separate objective function than the multi-objective optimization scheme. This means that the objective function vectors found by each single optimization scheme are not dominated by other objective function vectors obtained by SPEA2, and can be added to the Pareto front found by SPEA2. Although the single objective optimization schemes can identify better results for each separate objective function, they need to be run several times separately. On the other hand, with one trial SPEA2 can find multiple objective function vectors that perform as well as the parameter solutions obtained by GA. These results are in agreement with the "no free lunch theorem" that states "for any optimization algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class" (Wolpert and Macready, 1997). For example, GA-sum achieved better objective

values for E_{ns} -Tolgate and E_{ns} -Sum, while SPEA2-Sum obtained better results for E_{ns} -Salmon, and E_{ns} -Outlet. For parameter calibration, the single objective and multi-objective optimization schemes can find parameter solutions that are not inferior to each other.

Test results show that a substantial difference exists between the objective function values at different monitoring sites obtained by each single objective optimization scheme. For example, GA-sal obtained an E_{ns} value of 0.854 at Salmon, while GA-tol obtained an E_{ns} value of -0.361 at Salmon. Optimizing the objective function value at one site can lead to a serious bias of objective function values at other sites. Similarly, the multi-objective optimization scheme can also obtain a wide range of objective function values at each monitoring site. For example, the range of objective function values at Salmon, Tolgate, Outlet are [0.097, 0.827], [0.02, 0.599], and [0.471, 0.763], respectively. Given the substantial variation of the optimized objective function values, the corresponding parameter values were expected to scatter within the feasible space. The normalized parameter values obtained by different optimization schemes are shown in Figure 4-5. All of the parameter values were normalized between their upper and lower bounds so that they ranged between 0 and 1. As expected, the value of each parameter varied substantially depending on the selected optimization scheme. The range of the initial CN was 45 to 60. For the single objective optimization schemes, the optimized parameter values varied from each other. For example, the normalized CN values are 0.99, 0.12, 0.54, and 0.67 for GA-sal, GA-tol, GA-out and GA-sum, respectively. For SPEA2, the range of CN values obtained by the 96 Pareto optimal parameter sets is [0.38, 0.99]. These differences between the optimized parameter values reveal the relationship between streamflow and topography, landuse, and precipitation are different for each subwatershed, which result in a specific parameter solution for a given subwatershed.

The eight representative parameter sets obtained by both single and multi-objective optimization schemes were used to simulate the hydrographs at Salmon, Tolgate, and Outlet (Figure 4-6). In Figure 4-6, thick solid lines are the observed hydrograph; thin

solid lines are the simulated hydrographs using eight representative parameter sets calibrated by different optimization schemes. Considerable variation among the optimization schemes is apparent in the simulated hydrograph for each of these stations (Figure 4-6). The two sample Kolmogorov-Smirnov test (Massey, 1951) was used to test whether there is a statistically significant difference between the simulated hydrographs using the parameter sets obtained by different optimization schemes (Table 4-2). Of the total 28 comparisons at each monitoring site, 21, 25, and 22 pairs of hydrographs are significantly different from each other. This indicates that the selection of parameter optimization schemes can lead to significantly different simulated hydrographs, which may yield important implications to water resources management investigations.

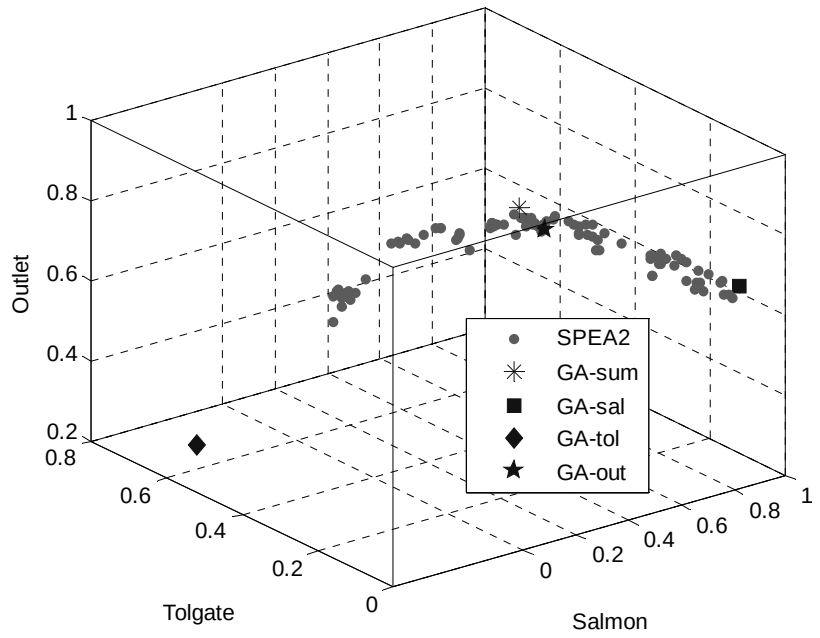

Figure 4-4. The best or Pareto optimal objective function values at Salmon, Tolgate and Outlet obtained by the GA-sal, GA-tol, GA-out, GA-sum, and SPEA2.

Table 4-1. The objective values obtained by eight parameter optimization schemes at Salmon, Tolgate, and Outlet for calibration period.

	Salmon	Tolgate	Outlet	Sum
GA-sal	0.854	0.022	0.690	1.566
GA-tol	-0.361	0.621	0.248	0.507
GA-out	0.499	0.291	0.776	1.566
GA-sum	0.615	0.441	0.741	1.797
SPEA2-sal	0.827	0.027	0.667	1.574
SPEA2-tol	0.132	0.599	0.471	1.201
SPEA2-out	0.548	0.181	0.763	1.492
SPEA2-sum	0.702	0.306	0.742	1.750

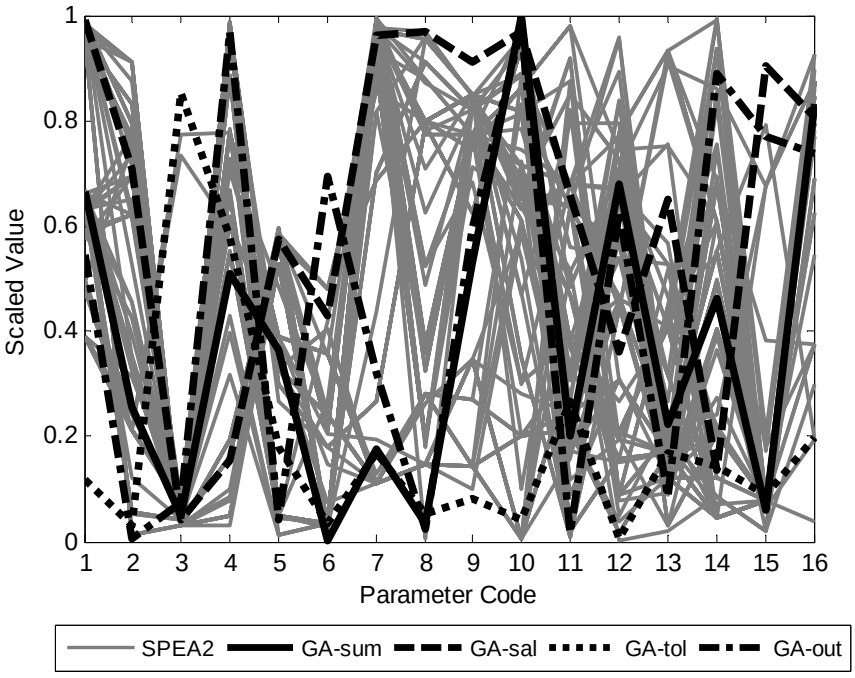

Figure 4-5. Normalized parameter values obtained by the SPEA2 and different GA based single objective optimization schemes.

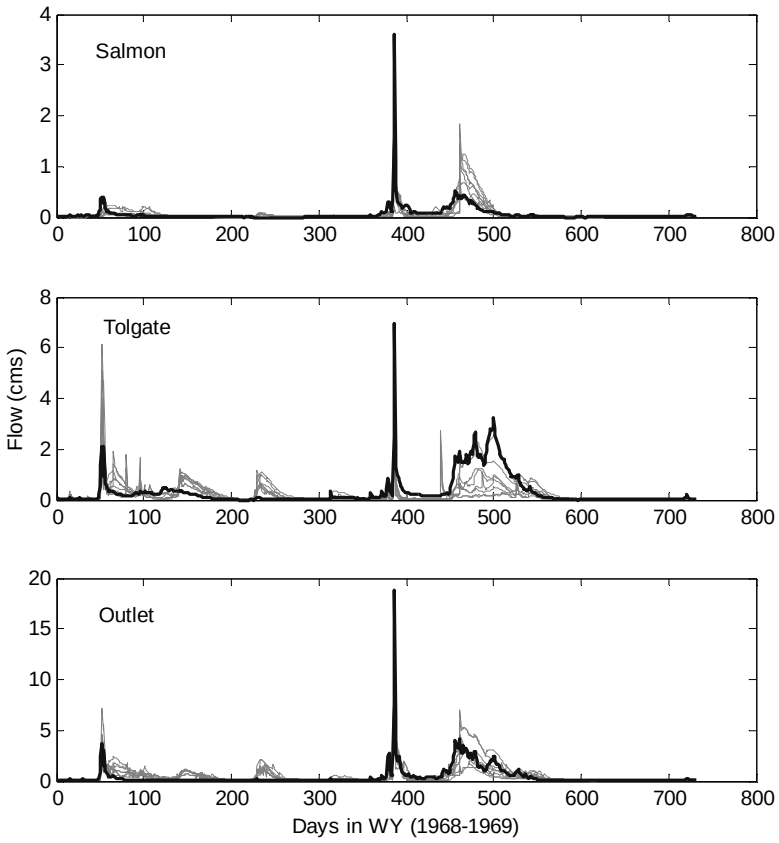

Figure 4-6. Simulated hydrographs using parameter sets calibrated by different optimization schemes for calibration period.

Table 4-2. Kolmogorov-Smirnov test results between the hydrographs simulated using different parameter solutions obtained by different optimization schemes at Salmon, Tolgate, and Outlet for calibration period.

		GA_sal	GA_tol	GA_out	GA_sum	SPEA2-sal	SPEA2-tol	SPEA2-out	SPEA2-sum
Salmon	GA_sal	--							
	GA_tol	1	--						
	GA_out	1	1	--					
	GA_sum	1	1	0	--				
	SPEA2-sal	0	1	1	1	--			
	SPEA2-tol	1	1	1	1	1	--		
	SPEA2-out	1	1	0	0	1	1	--	
	SPEA2-sum	1	1	0	1	0	1	0	--
Tolgate	GA_sal	--							
	GA_tol	1	--						
	GA_out	1	1	--					
	GA_sum	1	1	0	--				
	SPEA2-sal	0	1	1	1	--			
	SPEA2-tol	1	0	1	1	1	--		
	SPEA2-out	1	1	1	1	1	1	--	
	SPEA2-sum	1	1	1	1	1	1	1	--
Outlet	GA_sal	--							
	GA_tol	1	--						
	GA_out	1	1	--					
	GA_sum	1	1	0	--				
	SPEA2-sal	0	1	1	1	--			
	SPEA2-tol	1	0	1	1	1	--		
	SPEA2-out	1	1	0	0	1	1	--	
	SPEA2-sum	1	1	0	1	1	1	1	--

Note: 1 represents that there is significant difference between the two simulated hydrographs at significant level of 0.05, while 0 means the inverse.

4.3.2 Model validation using the parameters obtained by different optimization schemes

After calibration, the optimized parameter sets need to be validated using another independent set of observed data. The observed daily streamflow (1970-1972) at Samlon, Tolgate, and Outlet were used to validate the optimized parameter sets obtained by different optimization schemes. Figure 4-7 shows the simulated and observed hydrographs at Salmon, Tolgate, and Outlet. The hydrographs simulated by different parameter sets show evident variation from each other for the validation period. In Figure 4-7, thick solid lines are the observed hydrograph; thin solid lines are the simulated hydrographs using eight representative parameter sets calibrated by different optimization schemes. Using the two sample Kolmogorov-Smirnov test, it is shown that there are 20, 26, and 24 pairs of hydrographs that are significantly different from each other at Salmon, Tolgate and Outlet, respectively (Table 4-4), which indicate that the selection of different parameter sets has significant influence on the simulation results for water resources management investigations.

The evaluation coefficients (Table 4-3) show that the GA-sum and SPEA2-sum performed much better for the validation than other parameter calibration schemes. If only one specific objective function was emphasized in the calibration process, the calibrated parameter sets tended to achieve relatively good performance for that specific objective function at the cost of the performance of the other objective functions. For example, GA-tol and SPEA2-tol achieved a value of E_{ns} -Tolgate larger than 0.25 while E_{ns} -salmon was less than -0.35. In addition, the emphasis on one specific objective function tends to calibrate the model parameters so that they fit that specific objective function, while ignoring important information contained in other objective functions. On the contrary, the GA-sum and SPEA2-sum consider all objective functions and search for a good compromise among these different objective functions. Given the uncertainties associated with the observed data and model structure, the over fit to one specific objective function leads to poor performance during the validation period. For example, GA-sum and SPEA2-sum achieved much better E_{ns} values at all three

monitoring sites than other optimization schemes that only emphasized on one specific objective function during calibration. Hence, thegood performance of GA-sum and SPEA2-sum stresses the importance of collecting more detailed spatially distributed data to calibrate the distributed hydrologic model.

Table 4-3. Ojective values obtained by different parameter optimization schemes at Salmon, Tolgate, and Outlet for the validation period.

	Salmon	Tolgate	Outlet	Sum
GA-sal	0.305	0.009	0.561	0.875
GA-tol	-0.480	0.353	0.430	0.304
GA-out	0.083	0.117	0.615	0.815
GA-sum	0.320	0.507	0.728	1.554
SPEA2-sal	0.288	0.012	0.586	0.886
SPEA2-tol	-0.374	0.256	0.445	0.327
SPEA2-out	0.199	0.241	0.630	1.070
SPEA2-sum	0.438	0.488	0.725	1.651

Often hydrologic conditions of a validation period are different from those of a calibration period, which may lead to differences in performance of the parameter solutions for the respective periods. For example, GA-sum achieved the highest E_{ns} values based on the sum of the three sites for the calibration period, but SPEA2-sum performed better than GA-sum for the validation period. The 92 Pareto optimal parameter sets (except for SPEA2-sal, SPEA2-tol, SPEA2-out and SPEA2-sum) achieved by SPEA2 were also evaluated for the validation period. Some of the parameter sets outperformed the eight representative parameter solutions obtained during calibration. The best validation objective function values are 0.467 at Salmon, 0.526 for Tolgate, 0.739 for Outlet, and 1.68 for the sum, respectively. These values are better than those listed in Table 4-3. Among the 96 Pareto optimal parameter sets, 22 of them

achieved the sum of three objective function values larger than 1.55, and two of them achieved better results than GA-sum for all three objective functions. These validation results show that the multiple Pareto optimal parameter sets obtained by SPEA2 may contain some useful information that GA did not identify. The multiple Pareto optimal parameter sets can also allow practitioners to use different ways to select reasonable parameter values. If only one "most likely" parameter set is used to forecast or simulate streamflow, graphical visualization techniques (Gupta et al., 1998) and expert knowledge (Khu and Madsen, 2005) can be applied to assist the parameter selection.

4.4 Summary

With the increasing availability of spatial hydrologic data and growing popularity of complex, physically-based distributed hydrologic models, the use of the spatial data to calibrate and validate hydrologic models is becoming an increasingly important issue. In this study, different optimization schemes were applied to optimize a distributed hydrologic model, SWAT, using observed streamflow data at three monitoring sites within the Reynolds Creek Experimental Watershed in Idaho. The results show that different optimization schemes can lead to substantially different objective function values, parameter solutions, and corresponding simulated hydrographs. This in turn indicates that the selection of optimization schemes can significantly impact how well hydrologic models simulate actual streamflow. Parameters estimated by optimizing the objective function at three monitoring sites consistently produced better goodness-of-fit than those obtained through optimizing the objective function at a single monitoring site, which stresses the importance of having spatially distributed data to conduct such simultaneous mult-site calibration. When applied with multi-site data, the single objective (GA) method can identify better parameter solutions in calibration period, but the the multi-objective (SPEA2) method performed better in the validation period. The multi-objective optimization method, however, can identify multiple Pareto optimal parameter solutions, which allows hydrologic practitioners to use expert knowledge and visual graphic analysis to select one preferred solution. The multi-objective optimization method also eliminates the multiple runs by determining the optimal values

simultaneously. Overall, the application of different optimization schemes in the Reynolds Creek Experimental Watershed showed that the single objective (GA) and multi-objective (SPEA2) optimization methods both produce reasonable results for multi-site calibration and validation of the SWAT model. We also agree with the "no free lunch theorem" (Wolpert and Macready, 1997). Each optimization scheme has its strengths and weaknesses and may perform better under one set of hydrologic conditions as compared to another; therefore a method to combine the strengths of different optimization schemes deserves further research in the future.

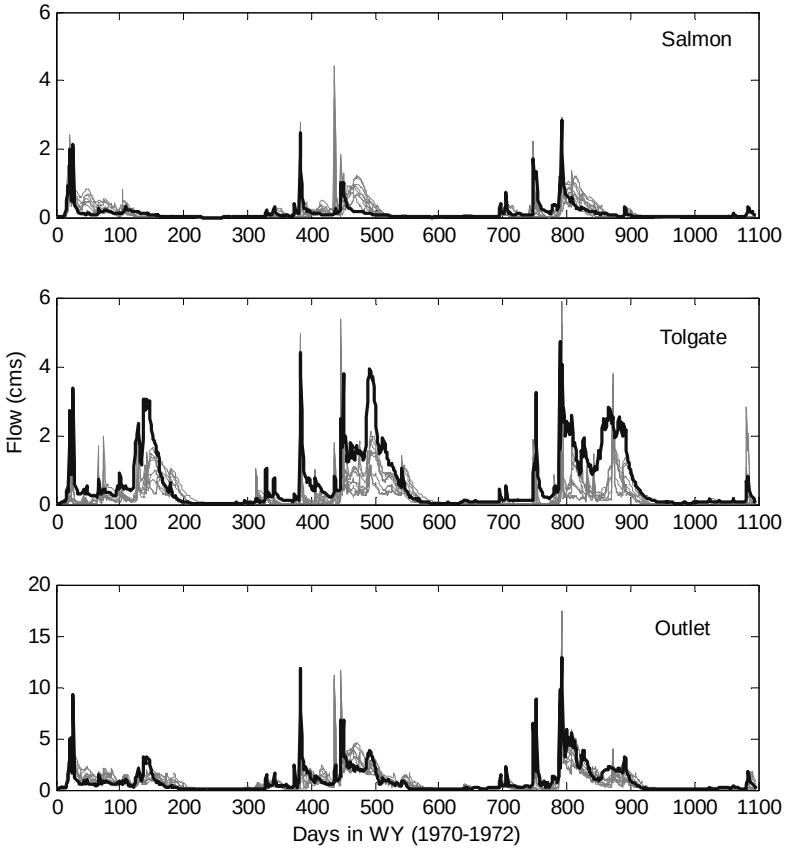

Figure 4-7. Simulated hydrographs using parameter sets calibrated by different optimization schemes for validation period.

Table 4-4. Kolmogorov-Smirnov test results between the hydrographs simulated using different parameter solutions obtained by different optimization schemes at Salmon, Tolgate, and Outlet for validation period.

		GA_sal	GA_tol	GA_out	GA_sum	SPEA2-sal	SPEA2-tol	SPEA2-out	SPEA2-sum
Salmon	GA_sal	--							
	GA_tol	1	--						
	GA_out	1	1	--					
	GA_sum	1	1	1	--				
	SPEA2-sal	1	1	0	0	--			
	SPEA2-tol	1	0	1	1	1	--		
	SPEA2-out	1	1	0	1	1	0	--	
	SPEA2-sum	1	1	0	0	0	1	1	--
Tolgate	GA_sal	--							
	GA_tol	1	--						
	GA_out	1	1	--					
	GA_sum	1	1	1	--				
	SPEA2-sal	1	1	1	1	--			
	SPEA2-tol	1	0	1	1	1	--		
	SPEA2-out	1	1	1	1	1	1	--	
	SPEA2-sum	1	1	1	1	0	1	1	--
Outlet	GA_sal	--							
	GA_tol	1	--						
	GA_out	1	1	--					
	GA_sum	1	1	1	--				
	SPEA2-sal	1	1	1	1	--			
	SPEA2-tol	1	0	1	1	0	--		
	SPEA2-out	1	1	1	1	1	1	--	
	SPEA2-sum	1	1	1	1	0	0	1	--

CHAPTER V

MULTI-OBJECTIVE OPTIMIZATION ALGORITHMS FOR SWAT

5.1 Introduction

In many real world optimization problem, there are multiple objectives need to be considered simultaneous. In practical calibration of hydrologic models, it was found that single-objective functions are often inadequate to properly measure all of the characteristics of the observed data deemed to be important (Vrugt et al., 2003a). When calibrating a hydrologic model, one or more objectives might be used to measure the agreement between the observed and simulated variables. With the various measurement data of different variables or at different locations, multiple objectives need to be used to evaluate the agreement between simulated and observed variables. Usually, different objectives prefer different parameter values (Gupta et al., 1998, Madsen, 2003, England et al., 2006). Numerous parameters sets (Pareto parameter sets) will be identified not to be inferior to others in terms of all objective values. Without additional information, it is not possible to distinguish any of the parameter set as being objectively better than any other parameter set (Gupta et al., 1998).

Simultaneously considering multiple objectives related to multiple hydrologic fluxes (e.g. surface flow and subsurface flow) at multiple sites have led to increasing research on applying and developing multi-objective optimization algorithms for hydrologic model calibration (e.g. Gupta et al., 1998, 1999; Yapo et al., 1998; Wagener et al., 2001; Madsen, 2003; Vrugt et al., 2003a; Vrugt et al., 2005). In Chapter IV, it was shown that the multi-objective optimization algorithm can find multiple parameter sets favoring different characteristics of the hydrologic modeling system, and provide insights into parameter uncertainty as well as the limitations of a model (Gupta et al., 1998). There are numerous multi-objective optimization algorithms available for implementing multi-

objective optimization of hydrologic model. The Non-dominated Sorted Genetic Algorithm II (NSGAII) (Deb et al., 2002) and Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Zitzler et al., 2002) have been widely applied for calibrating parameters of hydrologic models. Several examples of newly developed multi-objective optimization algorithms for calibrating hydrologic model are: Yapo et al. (1998) and Gupta et al. (1998) extended SCE-UA to address multi-objective functions in the multi-objective complex evolution (MOCOM-UA) algorithm, which was further improved to a multi-objective shuffled complex evolution Metropolis (MOSCEM-UA) by Vrugt et al. (2003a); Gill et al. (2006b) combine the PSO algorithm for multi-objective optimization of SAC-SMA Model and Support Vector Machine; Reed et al. (2003) developed Epsilon Dominance Non-dominated Sorted Genetic Algorithm II (ε-NSGAII) and applied it for a four-objective groundwater monitoring application.

Although the multi-objective optimization algorithms have been widely applied in hydrologic model calibration, a majority of these studies focused on conceptual rainfall-runoff applications. Recently, with the popularity of physically based, distributed hydrologic models for understanding complex hydrologic processes, there are increasing number of studies focusing on developing multi-objective calibration strategies for distributed hydrologic models (Madsen, 2003; Ajami et al., 2004; Vrugt et al., 2005; Bekele, 2007). The distributed hydrologic models often have more complex structures and significantly larger parameter sets that need to be specified (Tang et al., 2006), which make the calibration of distributed hydrologic model a challenging problem. In addition, the distributed hydrologic models are often computationally intensive due to the detailed processes they are intended to simulate. The implementation of automatic calibration may take several days, weeks, or even months to finish one trial. The enormous amount of computational time severely constrains the effectiveness of automatic calibration. The increasing size and complexity of calibration problems being considered within the water resources literature necessitate rapid and reliable search (Tang et al., 2006).

Among all the available multi-objective optimization algorithms, comparing their performances on different problems is useful for future selection of algorithms for specific application. Several studies have compared the efficiency and effectiveness of different algorithms on calibrating hydrologic models. For example, Kollat and Reed (2005b) compared the performance of NSGAII, ε-NSGAII, Epsilon Dominance Multi-objective Evolutionary Algorithm (ε-MOEA), and SPEA2 on a four-objective long-term groundwater monitoring (LTM) design test case, and the results revealed that ε-NSGAII was superior to the other three methods. Tang et al. (2006) compared the performance of ε-NSGAII, MOSCEM-UA, and SPEA2 on two real world hydrologic modeling cases, and concluded that SPEA2 and ε-NSGAII attained superior results than MOSCEM-UA and SPEA2 was superior or competitive to ε-NSGAII.

The purpose of this study is to assess the efficiency and effectiveness of several state-of-the-art multi-objective optimization algorithms for parameter calibration of SWAT. The strength and weakness of each algorithm were evaluated and discussed. A new multi-objective optimization algorithm was proposed and compared with other methods. The state-of-the-art multi-objective optimization algorithms that were tested in this study include SPEA2, NSGAII, ε-NSGAII, multi-objective particle swarm optimization (MOPSO-IEEE) (Coello Coello et al., 2004), MOPSO-EM (Reddy and Kumar, 2007), and MOPSO-WRR (Gill et al., 2006b). These algorithms were applied for calibrating parameter of SWAT in several study areas, and the performances of different algorithms were derived based on several evaluation coefficients. Finally, suggestions on the selection of multi-objective optimization algorithms for calibrating SWAT were provided.

5.2 Description of multi-objective optimization algorithms

Before the description of the multi-objective optimization algorithms, several common variables are firstly introduced here: A is non-dominated set; P_t represents the population for evolution, and $\overline{P}_t$ denotes the archive saving the elitist parameter sets. Other common variables used in the following sections are the same as those described

in Chapters III and IV. The Strength Pareto Evolutionary Algorithm 2 (SPEA2) (Zitzler et al., 2002) was introduced in Chapter IV.

5.2.1 NSGA-II

NSGA-II is an elitist multi-objective GA developed by Deb et al. (2002). The NSGAII algorithm has been used to allocate optimal waste load in rivers (Yandamuri and Srinivasan et al., 2006), multi-objective optimization problem considering minimizing the total design cost and robustness (Kapelan and Savic et al. 2005), and watershed water quality management problem considered meeting water quality targets while sustaining necessary growth (Dorn and Ranjithan, 2003). The major steps to implement the NSGAII are listed blow (Deb et al., 2002):

1) **Initialization:** Generate an initial population P_0 and evaluate each individual's fitness (if $t = 0$), otherwise use the population inherited from previous iteration. The GA algorithm is used to create a child population Q_t of size N.
2) **Combine parent and child population:** $R_t = P_t \cup Q_t$
3) **Fast non-dominated sorting of** R_t.
4) **Select the best individuals as parent population for generating new individuals.** Select N individuals from R_t into P_{t+1} using crowd distance sorting and crowded comparison operator.
5) **Termination:** If $t > T$ or another stopping criterion is satisfied then set A to the set of decision vectors represented by the non-dominated individuals. Stop.
6) Go to step 1).

5.2.1.1 Fast non-dominated sorting approach

First, for each individual we calculate two entities: (i) n_i, the number of solutions which dominate the individual i, and (ii) S_i, a set of individuals which the solution i dominates. The individuals with $n_i = 0$ are put in a list F_1, which is called the current

front. For each individual in the current front, the dominated individuals in S_i are identified. For each j in S_i, its n_j value is reduced by 1. The individuals (with $n_j = 0$) will be put to a new list F_2, and the current front is set to be F_2. The same processing procedures will be continued until all individuals in the population are assigned to a specific front.

5.2.1.2 Density estimation

In the fast non-dominated sorting approach, usually many individuals located in the same front. Then density estimation of each individual is used to discriminate the individuals with same front order. NSGAII use the average distance ($i_{distance}$) of the two individuals on either side of individual i along each of the objectives as an estimate of the size of the largest cuboid enclosing the point i without including any other point in the population (this distance is called crowding distance). The crowding distance of ith individual in its front is the average side-length of the cuboid shown with a dashed box in Figure 5-1.

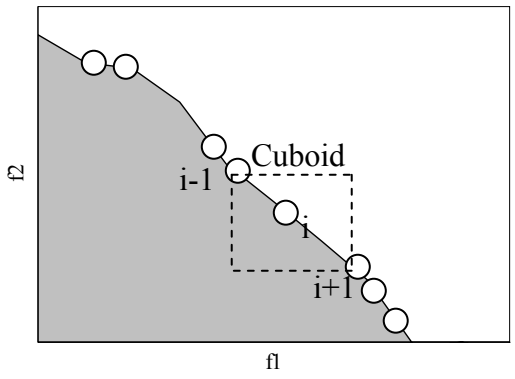

Figure 5-1. Illustration of crowding distance calculation (Modified from Deb et al., 2002).

5.2.1.3 Crowded comparison operator

After calculation of the non-domination front rank (i_{rank}) , and local crowding distance ($i_{dis\tan ce}$). The crowded comparison operator ($\geq_n$) was developed by Deb et al. (2002) for guiding the selection of the most appropriate individuals into the external archive. The $\geq_n$ operator is defined as

$$i \geq_n j \quad \forall (i_{rank} < j_{rank}) \vee (i_{rank} < j_{rank}) \wedge (i_{dis\tan ce} > j_{dis\tan ce}) \qquad \textbf{5- 1}$$

5.2.2 ε-NSGAII

ε-NSGAII is a relatively new multi-objective optimization algorihtm, which incorporates ε-dominance archiving (Laumanns et al., 2002) and automatic parameterization (Reed et al., 2003) to improve the original NSGAII's efficiency, reliability and ease-of-use (Kollat and Reed, 2005b). The ε-Dominance concept is introduced by Laumanns et al. (2002) for improving convergence and diversity in evolutionary multi-objective optimization. The basic concept of ε-Dominance is introduced below following Laumanns et al. (2002):

Definition 1 (ε-Dominance) If $\mathbf{x}_1$ is said to ε-dominates $\mathbf{x}_2$ for some ε>0, denoted as $\mathbf{x}_1 \succ_\varepsilon \mathbf{x}_2$, if and only if $\forall i \in \{1,\ldots m\}, (1+\varepsilon) \times f_i(\mathbf{x}_1) \geq f_i(\mathbf{x}_2)$

Definition 2 (ε-approximate set) Let $F \subseteq R^m$ be a set of vectors and ε>0. Then a set F_ε is called an ε-approximate Pareto set of F , if any vector $\mathbf{x}_1 \in F$ is ε-dominated by at least one vector $\mathbf{x}_2 \in F$, i.e. $\forall \mathbf{x}_1 \in F : \exists \mathbf{x}_2 \in F_\varepsilon$ such that $\mathbf{x}_1 \succ_\varepsilon \mathbf{x}_2$. The set of all ε-approximate Pareto sets of F is denoted as $P_\varepsilon(F)$.

Definition 3 (ε-Pareto set) Let $F \subseteq R^m$ be a set of vectors and ε>0. Then a set $F_\varepsilon^* \subseteq F$ is called an ε-Pareto set of F if i) F_ε^* is an ε-approximate Pareto set of F , i.e., $F_\varepsilon^* \in P_\varepsilon(F)$, and ii) F_ε^* contains Pareto points of F only, i.e., $F_\varepsilon^* \subseteq F^*$. The set of all ε-Pareto sets is denoted as $P_\varepsilon^*(F)$.

One practical method to maintain ε-Pareto set is using ε-grid, i.e. maintain a set of non-dominated boxes (one solution per box). When a new solution is generated, if and

only if i) its corresponding box is not dominated by any box represented by the archive A and ii) any other archive member in the same box is dominated by the new solution, then the new solution is added to the Pareto front. The employing of the ε-Pareto set concept is helpful for improving coverage (diversity) of non-dominated solutions (Laumanns et al., 2002). Knowles and Corne (2000) and Coello Coello (2004) presented a practical way to apply ε-Pareto set concept to keep diversity of the non-dominated solutions in objective function space. The objective function space of the solutions in external archive is divided into regions as shown in Figure 5-2, a two objectives case. For the grid with two or more solutions located in, Coello Coello (2004) randomly chose one of them and removes the others. Another method is to using crowding distance calculation (introduced in previous sections) to estimate the density solutions located in the same grid box, and choose the one with highest crowding distance. The results of implementation of the ε-Pareto set based on crowding distance are illustrated in Figure 5-2. In the process of implementation of the ε-Pareto set, if the individual inserted into the external archive lies outside the current bounds of the grid, then the grid has to be recalculated. The adaptive grid is really a space formed by hypercubes for optimization problems with dimensions equal to the number of objective functions that are needed to be optimized.

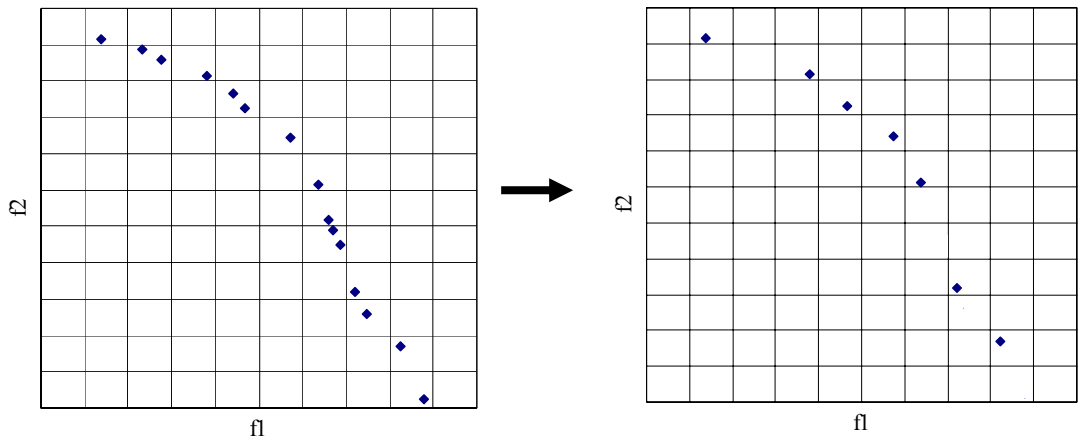

Figure 5-2. Illustration of implementation of the ε-Pareto set. (Modified from Coello Coello, 2004).

Kollat and Reed (2005a) incorporated the adaptive population sizing scheme into ε-NSGAII based on the population sizing theory and the automatic parameterization. In ε-NSGAII there are two types of iterations within the ε-NSGAII. The inter iteration shows how the population size is adapted. The intra iteration (NSGAII) is used to exploit and explore the ε-Pareto set of a given population. The ε-NSGAII uses a series of "connected runs" where small populations are exploited to pre-condition search with successively adapting population sizes (Kollat and Reed, 2005a). The major steps in ε-NSGAII are:

1) Initially, a population of small number is generated and evolved using original NSGAII until it is no longer making significant progress (intra iteration stop criterion).
2) Choose the ε-Pareto set from the previous NSGAII run and randomly generated new solutions (three times of the ε-Pareto set) to form a new population, which will be used for next implementation of NSGAII until it is no longer making significant progress
3) Repeat steps 2 until some inter iteration termination criteria are met.

The intra run will stop if 1) search within *n* generations (termed the lag window) fails to yield a specified percentage increase in the number of archived solutions or (2) the maximum run duration has been reached. After running the NSGAII for *n* generation, the ε-Pareto set size will be compared with that obtained *n* generations before, if the change percent of the ε-Pareto set size is larger than Δ (usually set to 10%) then the intra run will be stopped, and the current ε-Pareto set will be combined with newly generated random solutions (three times of the ε-Pareto set size) to form the population for next intra run using NSGAII. The inter run will stop if 1) maximum model evaluations are reached, or 2) maximum inter run number is reached.

5.2.3 MOPSO

Besides GA, PSO was also extended to handle multiple objectives problems. Particle swarm optimization (PSO) was first extended to deal with multiple objectives optimization by Coello Coello et al. (2004), who developed Multi-objective Particle Swarm Optimization algorithm (MOSPO). Coello Coello et al. (2004) compared the

MOPSO with NSGAII, Pareto Archive Evolutionary Strategy (PAES) and microgenetic algorithm for multi-objective optimization (microGA) on several standard multi-objective test problems, and the results showed that MOPSO is a competitive alternative of other popular algorithms. Reyes Sierra and Coello Coello (2006) conducted a survey of the state-of-the-art of the multi-objective particle swarm optimizers. Over twenty five different proposals of MOPSO were reported in published literatures. It is difficult to evaluate all these algorithms for the computationally intensive SWAT model. In this study, one MOPSO proposed by Reyes Sierra and Coello Coello (2005), which have been proved to be superior to several other PSO-based approaches, was selected as a benchmark MOPSO. This algorithm is referred as MOPSO-IEEE in this study. And two relatively new PSO-based algorithms that have been developed recently and successfully applied in water resources related multi-objective problems were chosen. One of them is elitist-mutation MOPSO (EM-MOPSO) (Reddy and Kumar, 2007), another was proposed by Gill et al. (2006b), which is denoted as MOPSO-WRR here. All the three algorithms are similar to each other in the general framework, but may vary in some specific procedures. The general framework of these three PSO-based algorithms are introduced below.

1) **Initialization.** Generate an initial population P_0 and create the empty archive (external repository) $\overline{P}_0 = \Theta$. Set the velocity of each particle in the population to 0. Set t = 0
2) **Fitness assignment.** calculate fitness values of individuals in P_t and $\overline{P}_t$
3) **Environmental selection.** copy all non-dominated individuals in P_t and $\overline{P}_t$ to $\overline{P}_{t+1}$. If size of $\overline{P}_{t+1}$ exceeds $\overline{N}$ then reduce $\overline{P}_{t+1}$ by means of the truncation operator,
4) **Update position of each particle.** Choose one particle in $\overline{P}_{t+1}$ as guide (Gbest), and generate new positions of the particles.
5) **Mutation.** The newly generated position is mutated.

6) **Termination.** $t = t + 1$. If $t > T$ or another stopping criterion is satisfied then set A to the set of decision vectors represented by the non-dominated individuals in $\overline{P}_{t+1}$. Stop.

7) Go to step 2).

The three PSO-based algorithms use the same "Initialization" and "Fitness assignment**"** operators, but may be different in the other operators. Table 5-1 lists the major characteristics of the four operators that have been adopted in different algorithms.

Table 5-1. Major characteristics of the three operators in different MOPSO algorithms.

	MOPSO-IEEE	MOPSO-EM	MOPSO-WRR
Environmental selection	ε-Pareto set	Pareto set with fixed size using crowding distance operator for truncation	Pareto set
Update position of each particle	Update particles in P_t. Gbest is randomly chosen from the ε-Pareto set	Update particles in P_t. Gbest is randomly chosen from the 10% less crowded Pareto set	Update particles in both P_t and $\overline{P}_t$. Choose the closest particle in the Pareto set as Gbest for particles in P_t. Choose the median of the Pareto set to guide the move of particles in $\overline{P}_t$.
Mutation	Dividing the P_{t+1} into three equal size part, and apply a variable range, fixed range, and no mutation to each part.	Elitist mutation	No mutation

5.2.3.1 Update position of particles in Pareto set

The operator to update position of each particle for the MOPSO-WRR is different from the other two. In MOPSO-WRR, it also generates new particles directly from the Pareto set. The position of each particle in the non-dominated set is updated with the guiding of the median particle of the Pareto set. Gill et al. (2006b) suggested the method to obtain the median particle of the Pareto set.

5.2.3.2 Mutation operators

There are three mutation operators. 1) In variable range operator, the range of the newly generated parameters are reduced with the increase of model evaluations; 2) In fixed range operator, the sampling range of the parameters will not change along the model evaluations; 3) The third mutation operator is EM range operator. This operator is relative complex. First, randomly select one of the objectives from m objectives. Second, sort the fitness function of particles in descending order and get the index number descending order sorted particles for the respective particles. Third, use crowding distance assignment operator and calculate the density of solutions in the Pareto set and sort them in descending order of crowding value. Finally, randomly select one of the least crowded solutions from the top 10% of Pareto set as guide, and Perform random mutation on a predefined number of particles (NM max).

5.2.4 A new multi-objective optimization algorithm

The test results of the above six multi-objective optimization algorithms show that PSO based methods converged quickly at the initial stage, while GA based methods could find better solutions than PSO based methods given large number of model runs. How to combine these two types of algorithms to produce more promising methods is an interesting topic. Actually the PSO and GA based methods are different in the update of solutions' position to find better solutions. PSO update the position of solutions in the population archive, while GA uses the promising solutions (elitists) to find the promising results. A new idea is generated based on the strong points of PSO and GA based methods. The solutions in the population are updated using PSO for the initial stage,

then the elitists (solutions in the non-dominated set) are updated using the GA algorithm. The basic procedures for the newly developed MO-PSOGA are introduced below:

1) **Initialization:** Generate an initial population P_0 and create the empty archive (external repository) $\overline{P}_0 = \Theta$. Set t = 0

2) **Fitness assignment:** calculate fitness values of individuals in P_t and $\overline{P}_t$

3) **Environmental selection:** copy all non-dominated individuals in P_t and $\overline{P}_t$ to $\overline{P}_{t+1}$. If size of $\overline{P}_{t+1}$ exceeds $\overline{N}$ then reduce $\overline{P}_{t+1}$ by means of the truncation operator.

4) **Termination:** if $t > T$ or another stopping criterion is satisfied then set A to the set of decision vectors represented by the non-dominated individuals in $\overline{P}_{t+1}$. Stop

5) **PSO operator:** Update the position of individuals in P_t using the PSO operator. The gbest solution is selected from $\overline{P}_{t+1}$. Evaluate each newly generated solution, update the pbest population, and reselect non-dominated solutions from pbest and $\overline{P}_{t+1}$. Set $\overline{P}_{t+1}$ to the newly generated non-dominated solutions. if t = 0, perform PSO until the non-dominated solutions do not increase 10% with 10 consecutive iterations. Otherwise, just perform PSO once.

6) **GA operator:** Perform crossover and mutation on the solutions in the $\overline{P}_{t+1}$. Store the newly generated solutions in a temporary archive P_{temp}. Evaluate each solution in P_{temp} Reselect non-dominated solutions from P_{temp} and $\overline{P}_{t+1}$. Set $\overline{P}_{t+1}$ to the newly generated non-dominated solutions. if t = 0, perform GA until the non-dominated solutions can not increase 10% with 10 consecutive iterations. Otherwise, just perform GA once.

7) Increment generation counter ($t = t + 1$) and go to step 3).

5.3 Performance of different multi-objective optimization algorithms

5.3.1 Evaluation metrics

Three issues are normally taken into account when assessing the performance of multi-objective algorithms (Zitzler et al., 2000):

1) Minimize the distance of the Pareto front produced by the algorithm with respect to the true Pareto front.
2) Maximize the spread of solutions found, so that we can have a distribution of vectors as smooth and uniform as possible.
3) Maximize the number of elements of the true Pareto optimal set found.

Many metrics for assessment the performance of different multi-objective optimization algorithms have been developed. In this study, five of them were employed. There are still many other metrics that can be used to evaluate performance of multi-objective algorithm. For further information please refer to Zitzler et al. (2003), Zitzler and Thiele (1999). The five metrics applied in this study are introduced below:

1) Generational distance (GD): The concept of generational distance was introduced by Van Veldhuizen and Lamont (1998), which estimates how far the elements are in the set of non-dominated vectors found so far from those in the true Pareto set. The equation to calculate GD is:

$$GD = \frac{\sqrt{\sum_{i=1}^{n} d_i}}{n} \qquad \textbf{5- 2}$$

Where n is the number of vectors in the set of non-dominated solutions found and d_i is the Euclidean distance (measured in objective space) between each of these and the nearest member of the Pareto set. A value of GD =0 indicates that all the elements generated are in the Pareto set. This issue from the list previously described.

2) Spacing (SP): The Spacing metric measures the range variance of neighboring vectors in the non-dominated vectors found (Deb, 2001)

$$SP = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(\bar{d} - d_i)^2} \qquad \textbf{5- 3}$$

where $d_i = \min_j(|f_1^i - f_1^j| + |f_2^i - f_2^j| + \ldots + |f_m^i - f_m^j|)$,

$i, j = 1, \ldots n$, $\bar{d}$ is the mean of d_i, and n is the number of vectors in the set of non-dominated solutions found. A value of $SP = 0$ indicates all members in the non-dominated set currently available are equidistantly spaced. This metric addresses the second issue from the list previously provided.

3) Error ration (ER): this metric was proposed by Van Veldhuizen and Lamont (1999) to indicate the percentage of solutions that are not members of the true Pareto optimal set:

$$ER = \frac{\sum_{i=1}^{n} e_i}{n} \qquad \textbf{5- 4}$$

Where n is the number of vectors in the set of non-dominated solutions. $e_i = 0$ if vector i is a member of the Pareto set, and $e_i = 1$, otherwise. A value of $ER = 0$ indicates all the vectors generated belong to the Pareto set. This metric addresses the third issue.

4) ε-indicator: this metric was proposed by Zitzler et al. (2003) to measure how well the algorithms converge to the true Pareto set or the best known approximation to the Pareto set. The ε-indicator represents the smallest distance that an approximation set must be translated to dominate the true Pareto set.

5) Hypervolume (HP): this metric was proposed by Zitzler and Thiele (1999). The HP metric measures how well the approximation set performs in identifying solutions along the full extent of the Pareto surface. The HP metric is represented by the difference between the volume of the objective space dominated by the Pareto set and the approximation set.

In general case, it is very difficult to find an analytical expression of the line or surface that contains all the Pareto optimal parameter sets. The normal procedure to generate Pareto front is to compute the feasible points Ω and their corresponding $\mathbf{f}(\Omega)$. When there are a sufficient number of feasible points, then the it is assumed the non-dominated points are approximating the Pareto front.

5.3.2 Test cases

Several test options could be applied to evaluate the performance of all these multi-objective algorithms for SWAT: 1) multi-site optimization (Calibrating stream flow, sediment or nutrients at different location simultaneously), 2) multi-variable optimization (calibrating different variables such as high flow, low flow, average flow), 3) multi-criteria optimization (optimizing different evaluation coefficients that measure the agreement between observed and simulated variables, such as coefficient of determination and Nash-Sutcliffe efficiency). As the time and computational resources are limited, the author will not test the algorithms for all the potential multi-objective optimization problems. In this study three test cases were designed to evaluate the performance of different algorithms.

1. Multi-criteria calibration of MCEW: the Nash-Sutcliffe efficiency and coefficient of determination for streamflow at the outlet (WE-38) are simultaneously optimized. The objective functions needed to be optimized are

$$F = \{f_1 = E_{ns}(WE-38), f_2 = R^2(WE-38)\} \quad \textbf{5- 5}$$

The calculation of the Nash-Sutcliffe efficiency has been described in Chapter III. Here, the formula used to calculate the coefficient of determination is:

$$R^2 = \left\{ \frac{\sum_{i=1}^{N}(Q_{obs,i} - \overline{Q}_{obs})(Q_{sim,i} - \overline{Q}_{sim})}{\left[\sum_{i=1}^{N}(Q_{obs,i} - \overline{Q}_{obs})^2\right]^{0.5}\left[\sum_{i=1}^{N}(Q_{sim,i} - \overline{Q}_{sim})^2\right]^{0.5}} \right\}^2 \quad \textbf{5- 6}$$

Where N is the total number of pairs of simulated and observed data, $i = 1,2,\cdots,N$, $Q_{obs,i}$ is observed flow on day i, $Q_{sim,i}$ is simulated flow on day i, $\overline{Q}_{obs}$ is the average of observed flow for the entire period, and $\overline{Q}_{sim}$ is the average of simulated flow for the entire period.

2. Multi-flow components calibration of MCEW: the Nash-Sutcliffe efficiency for three types of flow component are simultaneously optimized at the outlet (WE-38). The objective functions needed to be optimized are

$$F = \{f_1 = E_{ns}\_total(WE-38), f_2 = E_{ns}\_base(WE-38), f_3 = E_{ns}\_flood(WE-38)\} \quad \textbf{5- 7}$$

Three objectives that measure the performance of model response to high glow, low flow and total flow are formulated. Firstly, the baseflow separation technique developed by Arnold et al., (1999) was used to separate the base flow from the observed daily streamflow hydrograph. Then each day's major driven flow component is determined by a percent criterion ($flow_{base} / flow_{total}$). If $flow_{base} / flow_{total}$ is larger than 50%, then it's assumed this day's flow is mainly driven by base flow, otherwise by flood. Using this methods, we can obtain the days whose flow drive by base flow ($days_{base}$) and the days whose flow drive by flood ($days_{floodl}$). After separating baseflow and calculating the $days_{base}$ and $days_{flood}$. The three objective functions measuring the model's performance to different flow component can be formulated:

$$\textbf{Total flow: } E_{ns}\_total = 1 - \frac{\sum_{i=1}^{N}(Q_{obs,i\_total} - Q_{sim,i\_total})^2}{\sum_{i=1}^{N}(Q_{obs,i\_total} - \overline{Q}_{obs\_total})^2} \quad \textbf{5- 8}$$

$$\textbf{Baseflow: } E_{ns}\_base = 1 - \frac{\sum_{i=1}^{N\_base}(Q_{obs,i\_base} - Q_{sim,i\_base})^2}{\sum_{i=1}^{N\_base}(Q_{obs,i\_base} - \overline{Q}_{obs\_base})^2} \quad \textbf{5- 9}$$

High flow: $$E_{ns\_flood} = 1 - \frac{\sum_{i=1}^{N\_flood} (Q_{obs,i\_flood} - Q_{sim,i\_flood})^2}{\sum_{i=1}^{N\_flood} (Q_{obs,i\_flood} - \overline{Q}_{obs\_flood})^2} \quad \textbf{5- 10}$$

where, N is the total number of pairs of simulated and observed data, $i = 1,2,\cdots,N$, N_base is the number of pairs of simulated and observed data on $days_{base}$, $i\_base = 1,2,\cdots,N\_base$, N_flood is the number of pairs of simulated and observed data on $days_{flood}$, $i\_flood = 1,2,\cdots,N\_flood$, $Q_{obs,i}$ is observed flow on day i, Q_{sim,i_total} is simulated flow on day i, $\overline{Q}_{obs\_total}$ is the average of observed flow for the entire period, Q_{obs,i_base} is the observed flow on day i_base, Q_{sim,i_base} is the simulated flow on day i_base, $\overline{Q}_{base}$ is the average observed flow on $days_{base}$, Q_{obs,i_flood} is the observed flow on day i_flood, Q_{sim,i_flood} is the simulated flow on day i_flood, and $\overline{Q}_{flood}$ is the average observed flow on $days_{flood}$.

3. Multi-site calibration of RCEW: the Nash-Sutcliffe efficiency for three streamflow monitoring site (Salmon, Tolgate and Outlet) were optimized simultaneously. The objective functions needed to be optimized are

$$F = (f_1 = E_{ns}\textbf{-salmon},\ f_2 = E_{ns}\textbf{-Tolgate},\ f_3 = E_{ns}\textbf{-Outlet}) \quad \textbf{5- 11}$$

5.4 Results and discussion

5.4.1 Settings of multi-objective optimization algorithms

How to set the control parameters of the multi-objective optimization algorithms is critical for their performance. The population size, crossover rate, mutation rate, and archive size are expected to determine the performance of different algorithms. Actually the parameter setting problem is a trial and error problem. But the trail and error method is not suitable for computationally expensive model, as the time and resources are limited for running models. So we refer to the previously studies that have applied the

same multi-objective optimization algorithms. Based on the results from Tang et al., (2006), Coello Coello et al. (2004), Gill et al. (2006b), Reddy and Kumar (2007), the parameter settings of different multi-objective optimization algorithms are listed in Table 5-2. For different algorithms, settings of the control parameters are introduced as follows.

The control parameter settings for MOPSO-EM are (Reddy and Kumar, 2007): *c1* is 1.0 and *c2* is 0.5; inertial weight *w* is 1, constriction coefficient χ is 0.9; the size of elitist-mutated particles is set to 20% of the population size, the value of P_{em} was set to 0.2; and the value of S_m decreases from 0.2 to 0.01 over the iterations. For MOPSO-WRR, *c1* and *c2* are set to 0.5, inertial weight w decrease from 0.9 to 0.01 with the number of iterations. For MOPSO-IEEE, c1 and c2 are randomly chosen from [1.5, 2.0]; inertial weight w is randomly chosen from [0.1, 0.5], mutation rate is 0.5. For ε-NSGAII, the initial population size is 10, the lag window size is 50 for the first iteration and 10 after that.

Table 5-2. Parameters set for different algorithms.

Algorithm	Population size	Archive Size	Crossover rate	Mutation rate
NSGAII	50 and 100	50 and 100	0.5	1/(parameter dimension)
SPEA2	50 and 100	50 and 100	0.5	1/(parameter dimension)
ε-NSGAII	Initially set to 10	Varying with the population size	0.5	1/(parameter dimension)
MOPSO-IEEE	50 and 100	50 and 100	N/A	1/(parameter dimension)
MOPSO-EM	50 and 100	50 and 100	N/A	N/A
MOPSO-WRR	50 and 100	50 and 100	N/A	N/A

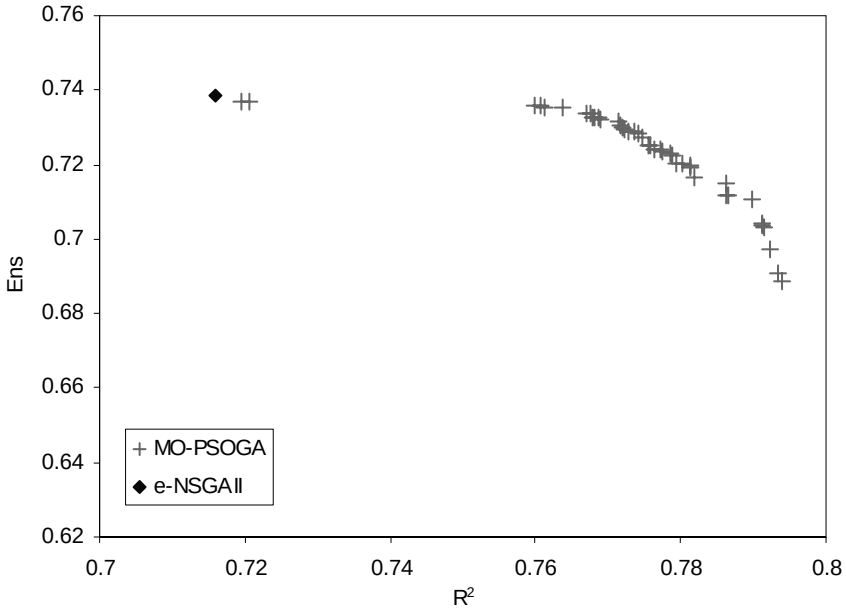

Figure 5-3. Pareto set found by all algorithms for the two-objective case in MCEW.

5.4.2 Evaluation of different algorithms for the two-objective case in MCEW

The best known approximation set was collected through running the seven multi-objective optimization algorithms. Only two algorithms (ε-NSGAII and the MO-PSOGA method) can contribute to the reference Pareto set (Figure 5-3). The ε-NSGAII contributed to 2% of the reference Pareto set, while the MO-PSOGA method contributed to the rest 98%. This indicates the advantage of combining the PSO and GA algorithms to perform multi-objective optimization of SWAT. The approximation set found by different algorithms are shown in Figure 5-4, and the evaluation coefficients of these approximation sets are listed in Table 5-3. The approximation set found by different algorithms refer to the best known non-dominated set found by all trials with different population sizes using the same multi-objective optimization algorithm.

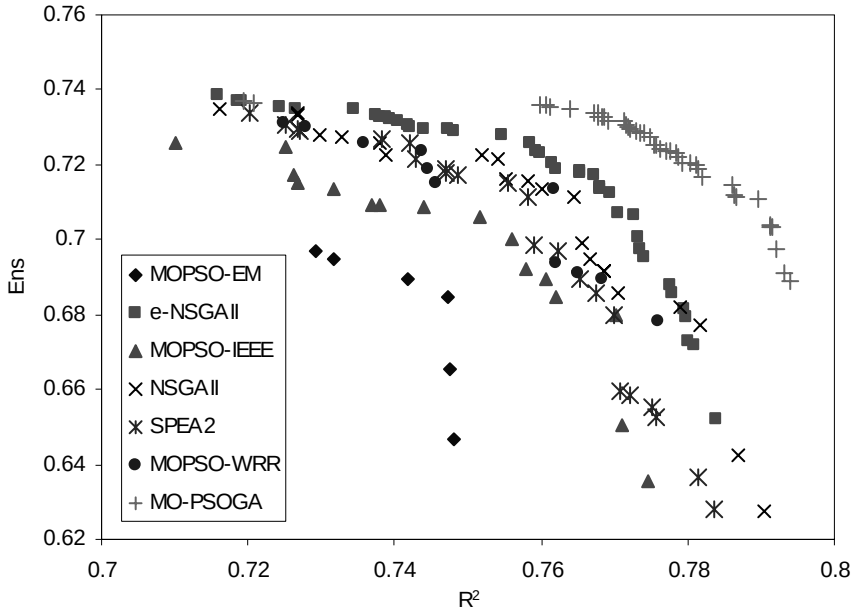

Figure 5-4. The approximation set found by different algorithms for the two-objective case in MCEW.

Table 5-3. Evaluation coefficients for the approximation set found by different algorithms for the two-objective case in MCEW.

Algorithms	GD	SP	ε-indicator	*ER*	HP
MO-PSOGA	0	0.0012	0.002	0.9767	1E-06
ε-NSGAII	0.0099	0.0035	0.017	0.0465	0.001
NSGAII	0.0129	0.0051	0.022	0	0.0015
SPEA2	0.0202	0.0029	0.025	0	0.0017
MOPSO-EM	0.0146	0.0066	0.046	0	0.0031
MOPSO-IEEE	0.0265	0.011	0.028	0	0.0023
MOPSO-WRR	0.0052	0.0056	0.024	0	0.0017

Through visual inspection of Figure 5-4, it is evident that the approximation sets found by different algorithms are different. Based on the evaluation coefficients in Table 5-3, the MO-PSOGA method that combining the PSO and GA algorithms performed the best in terms of GD, ε-indicator, ERR and HP. This reveals the promising ability of the newly proposed multi-objective optimization for parameter calibration of SWAT model. The ε-NSGAII method performed the second in terms of GD, ε-indicator, ERR and HP, which shows the advantage of dynamic population size. Among the rest five algorithms, the performances of NSGAII, SPEA2, and MOPSO-WRR are close to each other, while MOPSO-EM and MOPSO-IEEE performed the least.

The dynamic performance plots for the ε-indicator versus model evaluations for the two-objective case in MCEW are shown in Table 5-4. It was found that the PSO based multi-objective methods performed better than the GA based methods with small number of model evaluations, while the GA based methods performed better than the PSO based methods with large number of model evaluations. For example, with 2000 model runs, all the PSO based method obtained ε-indicator values close to or smaller than those obtained by GA based methods. But, for 10000 model evaluations, the ε-indicator values obtained by GA based methods values are close to or smaller than those PSO based methods. The evaluation results show that the PSO based methods tends to find relatively better parameter sets using less computational time, while GA based methods can obtained better parameter sets with larger number of model runs. This phenomenon is consistent with what we have found with the single objective optimization test cases. The implementation of the MO-PSOGA method that combining PSO and GA, the PSO is preferred to be run in the initial iterations, while GA is preferred to be run after this initial period. From Table 5-4, the population size also exerted appreciable effect on the performances of different multi-objective optimization algorithms, but there is no explicit rules that can be derived based on the test results for the two-objective case in MCEW. Among the three tested PSO based methods, MOPSO-WRR performed better than the other two. The smaller population size was chosen for

the PSO population size in the MO-PSOGA method because the MOPSO-WRR performed relatively better with small population size.

Table 5-4. Average ε-indicator value at different model evaluations for the two-objective case in MCEW.

Algorithms		2000	4000	6000	8000	1000
MO-PSOGA		0.075	0.0483	0.0333	0.0292	0.0272
ε-NSGAII		0.0847	0.045	0.0402	0.0338	0.031
NSGAII	50	0.0918	0.0504	0.039	0.0308	0.028
	100	0.1293	0.0597	0.0453	0.0363	0.033
SPEA2	50	0.1182	0.0655	0.0503	0.0425	0.035
	100	0.1422	0.071	0.0512	0.0428	0.0375
MOPSO-EM	50	0.0722	0.06	0.0553	0.0527	0.0523
	100	0.0788	0.0627	0.0583	0.0568	0.0545
MOPSO-IEEE	50	0.0708	0.0555	0.0502	0.0462	0.044
	100	0.0798	0.0542	0.05	0.045	0.0412
MOPSO-WRR	50	0.096	0.05	0.0423	0.0352	0.0342
	100	0.0882	0.067	0.0643	0.059	0.0542

5.4.3 Evaluation of different algorithms for the three-objective case in MCEW

As the time consumed by running the SWAT model is huge and the performances of MOPSO-EM and MOPSO-IEEE are the least among the seven algorithm as discussed in section 5.4.2, these two algorithms were not tested for the three-objective case in MCEW in order to save computation resources. The best known approximation set (Figure 5-5) was collected through running five multi-objective optimization algorithms. 57% of the estimated Pareto set was contributed by the newly proposed method, 40% by SPEA2, 2% by NSGAII and 1% by ε-NSGAII. The MOSPO-WRR contributed 0% for the Pareto set. The approximation set found by different algorithms are shown in Figure 5-6, and the evaluation coefficients of these approximation sets are listed in Table 5-5.

Through visual inspection of Figure 5-6, it is evident that the approximation sets found by different algorithms have different shape and extent.

The MO-PSOGA method performed the best in terms of GD, ε-indicator, ERR and HP, followed by SPEA2, NSGAII, ε-NSGAII and MOPSO-WRR. The good performance of the newly proposed method emphasizes its potential to be a promising multi-objective optimization algorithm. The performance rank of the five algorithms is different from that obtained in the two-objective case in MCEW. For example, ε-NSGAII performed less than SPEA2 and NSGAII for the three-objective case in MCEW. This reveals that the performances of different algorithms are influenced by the properties of some specific optimization problems.

The dynamic performance plots for the ε-indicator versus model evaluations for the three-objective case in MCEW are shown in Table 5-6. It is found that different algorithms show evidently different performance. Although the MO-PSOGA method obtained smaller ε-indicator value than SPEA2 with 10000 model evaluations, SPEA2 found better ε-indicator values with relatively small number of model evaluations. The average performances of SPEA2 and NSGAII are better with small population size.

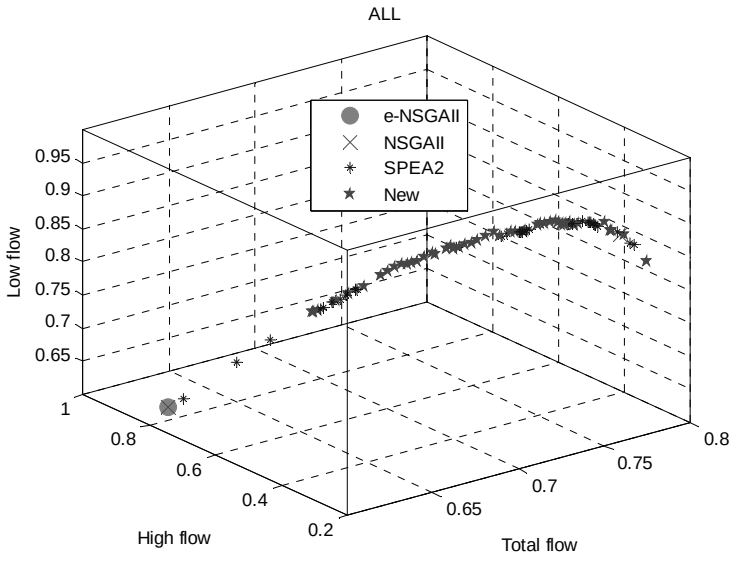

Figure 5-5. Pareto set found by all algorithms for the three-objective case in MCEW.

Table 5-5. Evaluation coefficients for the approximation sets found by different algorithms for the three-objective case in MCEW.

Algorithms	GD	SP	ε-indicator	*ER*	HP
MO-PSOGA	0.0043	0.004	0.011	0.6044	0.0001
ε-NSGAII	0.6063	0.0166	0.053	0.011	0.0068
NSGAII	0.0297	0.0121	0.038	0.033	0.0023
SPEA2	0.0074	0.0154	0.012	0.4615	0.0006
MOPSO-WRR	0.4134	0.1446	0.088	0	0.0079

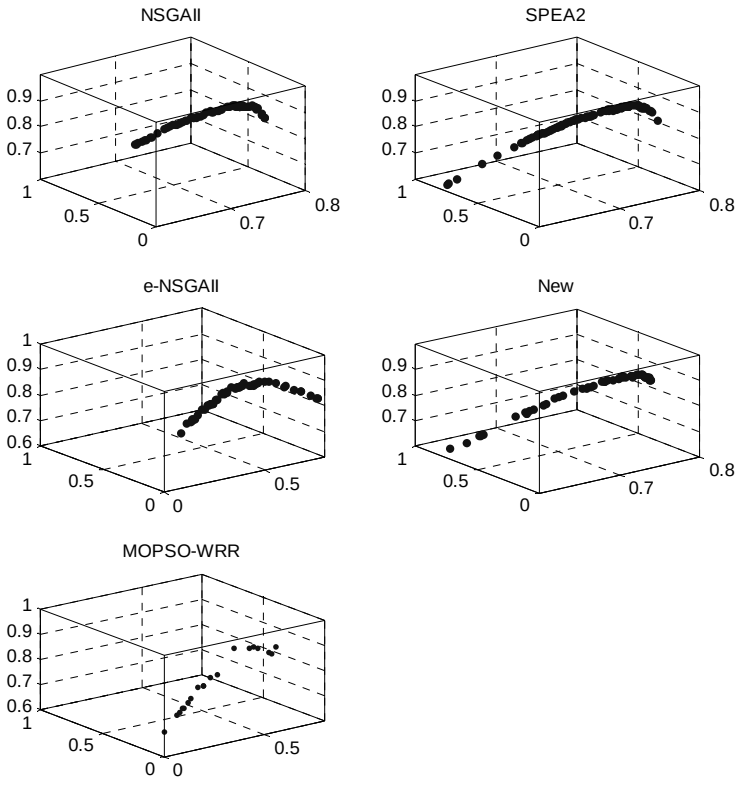

Figure 5-6. Approximation sets found by different algorithms for the three-objective case in MCEW.

Table 5-6. Average ε-indicator value at different model evaluations for the three-objective case in MCEW.

Algorithms		2000	4000	6000	8000	10000
MO-PSOGA		0.0818	0.0468	0.0404	0.0368	0.03
e-NSGAII		0.1218	0.1102	0.1054	0.1	0.0968
NSGAII	50	0.1083	0.0827	0.0777	0.063	0.052
	100	0.0948	0.0758	0.064	0.0618	0.058
SPEA2	50	0.0716	0.0434	0.0356	0.0336	0.0302
	100	0.1003	0.0753	0.0573	0.0418	0.0405

5.4.4 Evaluation of different algorithms for the three-objective case in RCEW

The last case is to test the performances of different algorithms on a three-objective multi-site calibration of the RCEW. In this case, in order to save computation resources, only three algorithms (MO-PSOGA, SPEA2, and NSGAII) were applied. The best known approximation set (Figure 5-7) was collected through running three multi-objective optimization algorithms. 39% of the estimated Pareto set was contributed by MO-PSOGA, 34% by SPEA2, and 27% by NSGAII. The approximation set found by different algorithms are shown in Figure 5-8, and the evaluation coefficients of these approximation sets are listed in Table 5-7. From Figure 5-8, the three algorithms still found different approximation sets with different shape and extent. The evaluation coefficients in Table 5-7 show that the performances of theses algorithms are very close to each other. The MO-PSOGA method performed slightly better than SPEA2 and NSGAII.

Table 5-7. Evaluation coefficients for the approximation set found by different algorithms for the three-objective case in RCEW.

Algorithms	GD	SP	e-indicator	*ER*	HP
MO-PSOGA	0.0018	0.0421	0.084	0.3913	0.0233
SPEA2	0	0.0234	0.094	0.337	0.0229
NSGAII	0	0.0419	0.096	0.2717	0.0447

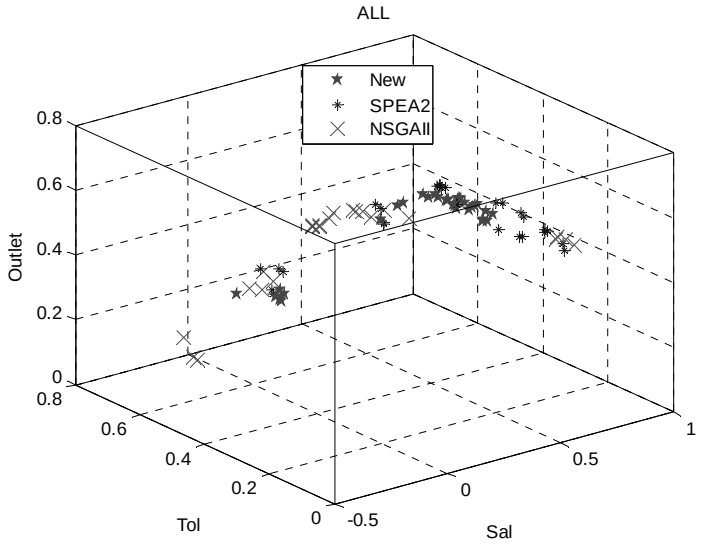

Figure 5-7. Pareto set found by all algorithms for the three-objective case in RCEW.

Table 5-8. Average ε-indicator value at different model evaluations for the three-objective case in RCEW.

Algorithms		2000	4000	6000	8000	1000
MO-PSOGA		0.204	0.1586	0.1452	0.1372	0.1156
NSGAII	50	0.1878	0.1604	0.1412	0.1288	0.1202
	100	0.2283	0.154	0.1373	0.129	0.1183
SPEA2	50	0.2168	0.1572	0.1364	0.1286	0.117
	100	0.2444	0.182	0.1522	0.1374	0.121

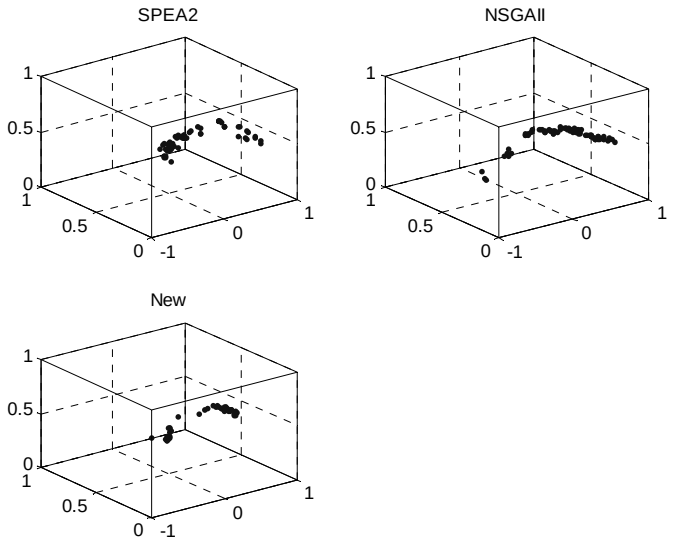

Figure 5-8. Approximation set found by all algorithms for the three-objective case in RCEW.

The dynamic performance plots for the ε-indicator versus model evaluations for the three-objective case in RCEW are shown in Table 5-8. Although the MO-PSOGA method obtained smaller ε-indicator value than SPEA2 with 10000 model evaluations, SPEA2 and NSGAII found better ε-indicator values with relatively small number of model evaluations.

5.5 Summary

In this chapter, six multi-objective algorithms and one newly proposed method (MO-PSOGA) were tested for parameter calibration of SWAT model for several multi-objective case studies. The application of GA based and PSO based algorithms show that the PSO based method converge quickly at the initial state, while GA based methods performed better in terms of finding good parameter sets with larger number of model runs. A new multi-objective optimization method (MO-PSOGA) that combines the advantages of the PSO and GA algorithms was proposed. Based on the evaluation of the performances of different algorithms on three test cases, the MO-PSOGA method consistently perform better for all the other algorithms in terms of finding good

parameter sets approximating the Pareto front with large number of runs. For relatively small number of evaluations of SWAT, the MO-PSOGA method can obtain results close to that of other optimization algorithms. Overall, the MO-PSOGA method can serve as a promising alternative method for multi-objective optimization of SWAT model.

CHAPTER VI

APPROXIMATING SWAT USING ANN AND SVM

6.1 Introduction

In recent years, complex, physically based hydrologic models have been used increasingly by hydrologists and resources managers as tools to understand and manage natural and human activities that affect watershed systems. With the popularity of these models, the time consumed for running these models is increasing substantially. The automatic calibration and uncertainty analysis require a large number of evaluations of the computationally intensive models. As the SWAT model usually requires several minutes, several hours, or even several days to be implemented for a single run, it is very time consuming to conduct the parameter calibration and uncertainty of SWAT. For example, the time consumed by running SWAT model 1,000 times is about 17 hours (if 1 minute is required for one run) or even 1000 hours (if 1 hour is required for one run). Because of the enormous computational cost involved, it is important to improve the efficiency of parameter calibration and uncertainty analysis for SWAT. How to produce good parameter solutions with a limited number of evaluations of SWAT is a concern of many model users.

The function approximation method is an efficient way for parameter calibration and uncertainty analysis of computationally intensive models (Gutmann, 2001). Several studies have applied different learning machines as surrogate models to approximate the behavior of computationally intensive environmental models. For example, the Artificial Neural Network (ANN) has been used by Morshed and Kaluarachchi (1998), Johnsom and Rogers (2000), Almasri and Kaluarachchi (2005), and Zou et al. (2007) as surrogates of complex environmental models for parameter selection and management practices evaluation. Radial Basis Function (RBF) also has been used by Mugunthan and

Shoemaker (2006) as an approximation tool of computationally expensive models for parameter calibration and uncertainty analysis. Previous studies have shown that different learning machines exhibited various abilities to approximate different models' behavior. For example, Khalil et al. (2005) compared the capacity of four learning machines (ANN, support vector machine (SVM), locally weighted projection regression (LWPR), and relevance vector machines (RVM)) for approximating the behavior of complex ground water quality models, and concluded that ANN minimizes empirical risk and SVM can minimize the structural risk to achieve estimators that are less susceptible to over fitting. To the best of the author's knowledge, the learning machines were seldom applied to approximate the SWAT model's response to parameter adjusting. Therefore, evaluating and comparing the performances of different learning machines for approximating the SWAT model is a topic deserving further exploration. Among the different learning machines, the ANN is very popular and has been successfully used to approximate the computationally intensive models. Another learning machine - SVM, which has exhibited learning ability equal to or better than ANN for hydrologic simulation (Liong and Sivapragasam, 2002; Gill et al., 2006), was also taken as a promising candidate learning machine. In this study, the major objective was to evaluate and compare the performances of ANN and SVM for approximating the response of SWAT model to parameter selections. Several practical issues related to efficient and effective application of the learning machines were also analyzed and discussed. The remainder of this paper is organized as follows: Section two provides a brief description of the Materials and Methodology, including the characteristics of the study area (LREW in GA and MCEW in PA), the description of the SWAT model, introduction of learning machines (ANN and SVM), and the procedures of implementing particle swarm optimization (PSO) algorithm used to train the learning machines. In section three, several test cases with different combinations of parameter dimensions, numbers of training sample points, and cross validation schemes were designed to evaluate and compare the performance of ANN and SVM. In section four, the results and discussion of the performance of ANN and SVM for approximating the SWAT model with respect

to different test cases were presented. In addition, the potential application learning machines for parameter calibration and uncertainty analysis of SWAT were also illustrated with simple analysis and example. Finally, a summary with conclusions is provided in section five.

6.2 Methods

6.2.1 ANN

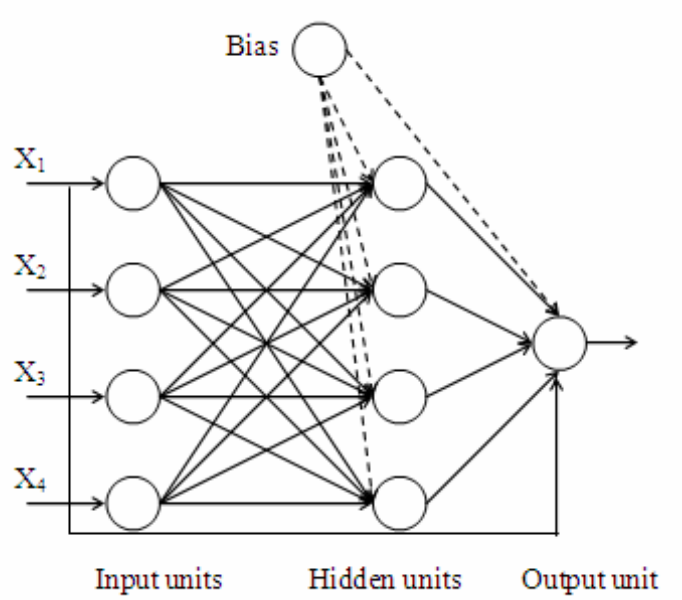

Figure 6-1. A fully connected one-hidden-layer feed-forward neural network with four input units, four hidden units, and one output unit.

ANN is a universal approximate that has been widely used to simulate complex and nonlinear relationships between input and output data. The input data vector $\mathbf{x}_t$ is mapped to the target variable $\mathbf{y}_t$ in the form of $\mathbf{y}_t = f(\mathbf{x}_t)$, where $f(\mathbf{x}_t)$ is the neural network function. ANN has been widely used in hydrology and water resources-related applications. An artificial neural network consists of an interconnected group of processing elements (artificial neurons) which can exhibit complex global behavior. One

typical type of a 3 layered feed-forward neural network with four inputs, four hidden units and one output is shown in Figure 6-1 (Liang, 2005). This network can be used to approximate the variable of interest using a function with the form of

$$f(\mathbf{x}_t) = \alpha_0 + \sum_{i=1}^{p} x_{it}\alpha_i + \sum_{j}^{M} \beta_j \psi(\gamma_{j0} + \sum_{i=1}^{p} x_{it}\gamma_{ji}) \qquad \textbf{6- 1}$$

where $\mathbf{x}_t$ is the input data vector at time t, p is the dimension of $\mathbf{x}_t$, x_{it} is the ith component of $\mathbf{x}_t$, M is the number of hidden units; α_0 denotes the bias of the output unit, α_i denotes the weight that directly connects the ith input unit to the output unit; β_j is the weight that connects the jth hidden unit to the output unit; γ_{j0} is the bias of the jth hidden unit, γ_{ji} denotes the weight on the connection from the ith input to the jth hidden unit; and $\psi(\cdot)$ is the activation function of the hidden units. The activation function applied in this study is the hyperbolic tangent function. The $\tanh(\cdot)$ function ensures that the output of a hidden unit is 0 if all connections to the hidden unit from input units have been eliminated. The symbols α_0, α_i, β_j, γ_{j0}, and γ_{ji} are the biases and connections that need to be optimized to infer an acceptable approximation of the relationship underlying a system that relates a set of input variables to the dependent variables of interest.

Given a set of l samples $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_L, y_L)\}$, where $\mathbf{x}_t$ are the input vectors and y_t are the corresponding output values ($t = 1,2,\ldots L$), a class of functions $f(\mathbf{x}_t)$ can be formulated to approximate the relationship between the input vector and the output variable. The optimal weights and biases of the links in the neural network are usually determined by empirical risk minimization (ERM) procedure. The formula used to calculate the empirical risk R_{emp} is

$$R_{emp} = \frac{1}{L}\sum_{t=1}^{L}(y_t - f(\mathbf{x}_t))^2 \qquad \textbf{6- 2}$$

The conventional method for solving the problem of regression estimation for a learning machine is to apply the ERM principle using the loss function specified by the

above equation, which is the least square method. There are many optimization algorithms available for training the ANN model. Herein, a popular evolutional optimization algorithm, Particle Swarm Optimization (PSO), was applied. The PSO has been successfully applied to optimize artificial neural networks for river stage prediction (Chau, 2006). For more information about PSO, please refer to Kennedy et al. (2001).

6.2.2 SVM

The Support Vector Machine developed by Vapnik (1998) is gaining wide use in many research fields. The basic form of SVM for nonlinear regression is

$$f(\mathbf{x}_t) = \mathbf{w}^T \cdot \phi(\mathbf{x}_t) + b \qquad \textbf{6- 3}$$

where $\mathbf{w}$ and b are the regression parameter vectors of the function. $\phi(\mathbf{x})$ is the nonlinear function that maps input data $\mathbf{x}_t$ into a feature space in which the training data may exhibit linearity. SVM employs the novel Structural Risk Minimization (SRM) principle developed from statistical learning theory to minimize the expected risk based on limited data (Vapnik 1998). The SRM principle suggests a tradeoff between the quality of the approximation and the complexity of the approximating function. A popular regression version of SVM, ε-SVM, is used to find a function that has at most ε deviations from the actual obtained targets for all the training data, and is as flat as possible (Smola and Scholköpf, 2004). The objective function is

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}\xi_t + C\sum_{i=1}^{l}\xi_t^* \qquad \textbf{6- 4}$$

$$\text{Subject to} \quad y_t - \mathbf{w}^T\phi(\mathbf{x}_t) - b \le \varepsilon + \xi_t,$$

$$\mathbf{w}^T\phi(\mathbf{x}_t) + b - y_t \le \varepsilon + \xi_t^*,$$

$$\xi_t, \xi_t^* \ge 0, t = 1,\ldots,l.$$

where ξ_t and ξ_t^* are slack variables that specify the upper and lower training errors subject to an error tolerance ε, and C is a positive constant that determines the degree of penalized loss when a training error occurs. The schematic illustration of the calculation of ξ_t and ξ_t^* is shown in Figure 6-2, where solid circles denote support

vectors. Only the points (x_i, y_i) outside the ε tube contribute to the loss function, and an error ξ_t or ξ_t^* needs to be calculated. ε-SVM avoids under-fitting and over-fitting the training data by minimizing both the regularization term $\frac{1}{2}\mathbf{w}^T\mathbf{w}$ and the training error term $C\sum_{i=1}^{l}\xi_t + C\sum_{i=1}^{l}\xi_t^*$. Minimizing the first term is equivalent to minimizing the complexity of the learning machine, and minimizing the second term corresponds to minimizing the empirical risk.

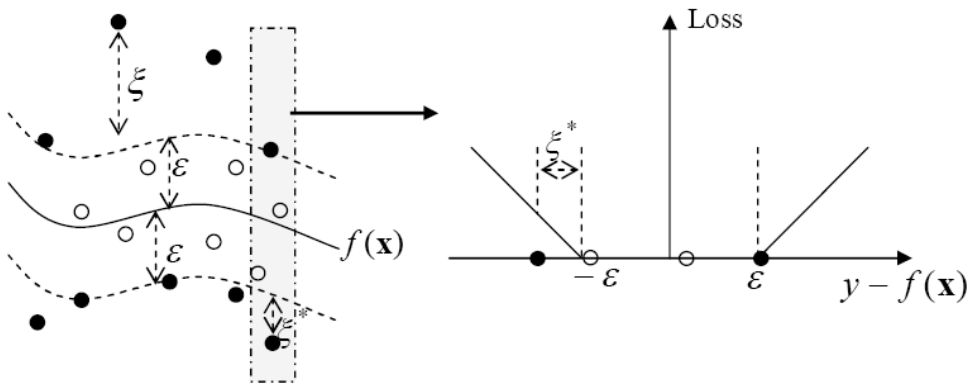

Figure 6-2. Nonlinear SVR with Vapnik's ε-insensitive loss function (Modified from Yu et al., 2006).

The optimization problem described above can be solved through introducing a dual set of Lagrange multipliers, α_t and $\overline{\alpha}_t$, enabling the optimization problem to be solved more easily in the dual form, by applying the standard quadratic programming algorithm. Then, the dual form of the nonlinear SVM optimization problem can be expressed as

$$\min_{\alpha_i,\overline{\alpha}_i} \frac{1}{2}\sum_{t,j=1}^{l}(\alpha_t - \overline{\alpha}_t)(\alpha_t - \overline{\alpha}_t)K(\mathbf{x}_t, \mathbf{x}_j) + \varepsilon\sum_{t=1}^{l}(\alpha_t + \overline{\alpha}_t) - \sum_{t=1}^{l} y_t(\alpha_t - \overline{\alpha}_t) \qquad \textbf{6-5}$$

$$\text{Subject to} \quad \sum_{t=1}^{l}(\alpha_t - \overline{\alpha}_t) = 0$$

$$0 \le \alpha_t \le C, t = 1,2,\ldots l$$

$$0 \le \overline{\alpha}_t \le C, t = 1,2,\ldots l$$

where $K(\mathbf{x}_t, \mathbf{x}_j)$ is $\langle \phi(\mathbf{x}_t), \phi(\mathbf{x}_j) \rangle$, the inner product of $\phi(\mathbf{x}_t)$ and $\phi(\mathbf{x}_j)$. The dual formulation does not require the explicit form of the nonlinear function $\phi(\mathbf{x})$. A "kernel" function $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_t), \phi(\mathbf{x}_j) \rangle$ is sufficient to perform the SVM optimization problem. The application of the kernel function avoids the direct computation of $\langle \phi(\mathbf{x}_t), \phi(\mathbf{x}_j) \rangle$, which may be too complex to perform. After the Lagrange multipliers, α_i and $\overline{\alpha}_i$, have been determined, the approximate function can be expressed as,

$$f(\mathbf{x}_t) = \sum_{k=1}^{l} (-\alpha_k + \overline{\alpha}_k) K(\mathbf{x}_t, \mathbf{x}_k) + b \qquad \textbf{6- 6}$$

If the values $(-\alpha_t + \overline{\alpha}_t)$ are zero, the corresponding data point are contained inside of the ε-intensive tube. Only those data points with non zero coefficients $(-\alpha_t + \overline{\alpha}_t)$ will be used in the final calculation of function (6-6). These data points are denoted as support vectors (SVs).

In order to calculate function (6-6), the kennel function $K(\mathbf{x}_t, \mathbf{x}_j)$ is necessary. Any function that meets Mercers condition can be used as a kernel function. The commonly used kernel functions include linear kernel, polynomial kernel, sigmoid kernel and radial basis function (RBF) kernel. According to the previous application of SVM (e.g., Yu et al., 2006; Gill et al., 2006a) and a preliminary test of the performance of different kernel functions, the RBF kernel was selected in this study. The basic form of the RBF kernel is,

$$K(\mathbf{x}_t, \mathbf{x}_j) = \exp(\gamma \| \mathbf{x}_t - \mathbf{x}_j \|^2]), \gamma > 0 \qquad \textbf{6- 7}$$

where γ is kernel parameter.

Based on previous descriptions of SVM, there are three important parameters that dominate the performance of the nonlinear SVR: the cost constant C, the radius of the insensitive tube ε, and the kernel parameter γ. These three parameters are usually determined by a heuristic trial-and-error process. Recently, advanced evolutionary

optimization algorithms have been proposed to calibrate the parameters of SVM (e.g., Bazi and Melgani, 2007). In this study, the PSO algorithm was applied for parameter selection of SVM. Based on previous studies on applying automatic optimization for SVM, the ranges of the C, γ, and ε were set to [10^{-3}, 100], [10^{-3},5], and [0, 0.5], respectively.

6.3 Test cases design

In this study, the optimization objective function is the Nash-Sutcliffe efficiency (E_{ns}) that has been introduced in Chapter III. The E_{ns} values in MCEW and LREW were calculated using six years (1995-2001) and four years (1997-2000) simulated daily streamflow, respectively. The ANN and SVM were applied to approximate the objective function (E_{ns}) values simulated by SWAT with different parameter values. The learning ability of ANN and SVM is dependent on many factors, such as number of training samples, parameter dimensions, training schemes (the number of folds of cross-validation), and characteristics of the watershed. In the following sections, several test cases were designed to evaluate the effect of the parameter dimensions, training schemes and training sample sizes on the approximation ability of ANNs and SVM in two watersheds (LREW and MCEW).

6.3.1 Parameter dimensions

In the previous application of SWAT for hydrologic modeling, the number of parameters that were calibrated usually ranges between five and sixteen parameters (Van Liew et al., 2003; Van Liew et al., 2007). Herein, four parameter dimension scenarios with six, nine, twelve and sixteen parameters, respectively, were designed for each watershed. In order to determine the parameters included into different parameters scenarios, all sixteen parameters identified by Van Liew et al. (2007) were ranked according to their sensitivity index in descending order. In the n-parameter scenario, the first n parameters are included. The sensitivity analysis method used in this study is the algorithm developed by van Griensven et al. (2006), which has been incorporated into the parameter sensitivity program of SWAT2005. The basic idea of LH-OAT is to perform Latin-hypercube (LH) sampling and then perform the One-factor-At-a-Time

(OAT) sensitivity analysis locally around each parameter point obtained in the LH procedure. Finally, the sensitivity indices obtained by the OAT operations are averaged over the number of Latin-hypercube samples to get the average estimate of the sensitivity of each parameter. Another efficient alternative to the Latin-hypercube design is symmetric Latin-hypercube design (SLHD). In the samples obtained by SLHD, each parameter point has its reflection through the center of the parameter space. The SLHDs have shown better performance than simple LH (Ye et al., 2000). In this study, the SLHD was applied to generate random initial sample parameter points. The parameter sensitivity ranks in LREW and MCEW are listed in Table 6-1.

Table 6-1. Sensitivity rank of the 16 parameters in the LREW and MCEW.

Sensitivity Rank	LREW	MCEW
1	CN	CN
2	Surlag	Surlag
3	ESCO	ESCO
4	GWQMN	SFTMP
5	CH_K2	TIMP
6	RCHRG_DP	SMTMP
7	SOL_AWC	SOL_AWC
8	GW_DELAY	GW_DELAY
9	ALPHA_BF	CH_K2
10	GW_REVAP	ALPHA_BF
11	REVAPMN	SMFMN
12	SFTMP	REVAPMN
13	SMTMP	GWQMN
14	SMFMX	RCHRG_DP
15	SMFMN	GW_REVAP
16	TIMP	SMFMX

6.3.2 Number of training samples

The effect of training sample size on the performance of ANN and SVM was investigated. As the function approximation method is designed for computationally expensive models, the maximum number of training samples should not be large. In this

study, the largest training sample size is 1000. The other sample sizes are equally spaced between 200 and 1000 with an increment of 200.

6.3.3 Training schemes for ANN and SVM

Cross-validation is one popular approach to estimating how well the model that has been trained is going to perform on another independent set of data. The cross validation approach requires the data set to be divided into *k* mutually disjointed folds (subsets) S_j ($\{j = 1,\ldots,k\}$). For each subset S_j, the function is trained based on all the data left. Totally, the training and testing operations are needed to be operated *k* times. The cross-validation with *k* mutually disjointed subsets of training samples are denoted as *k*-fold cross-validation. Previous studies have shown that 10-fold cross validation is preferred in real world problems (Kohavi, 1995). The computation time required by the cross validation scheme increases with the number *k*. As saving computation time is one of the major concerns, the optimal number of *k* will be examined through evaluating the performance of 10-fold, 5-fold, 4-fold, 3-fold, 2-fold and no cross-validation schemes. The objective function used in the *k*-fold cross validation is

$$Error = \frac{1}{k} \sum_{j=1}^{k} R_{emp,j} \qquad \textbf{6- 8}$$

where $R_{emp,j}$ is the tested mean square error for subset S_j. The *Error* was taken as an objective function to be minimized using the PSO algorithm. For the training of ANN, the appropriate number of hidden units (ranging from 5 to 50) was selected through cross-validation scheme. As the number of parameters that need to be optimized for ANN is many more than that for SVM, the population size of PSO used for ANN is 200, while the population size of PSO for SVM is 30. The criterion for stopping the optimization process is that if the optimization objective function does not improve by 2% within three consecutive iterations of PSO.

6.3.4 Evaluation of the performance of ANN and SVM

After the training of the ANNs and SVM, their generalization ability is evaluated using another independent input data set and corresponding target values. The evaluation coefficient used in this study was coefficient of determination (R^2). R^2 is calculated as:

$$R^2 = \left\{ \frac{\sum_{i=1}^{N}(O_i - \overline{O})(P_i - \overline{P})}{\left[\sum_{i=1}^{N}(O_i - \overline{O})^2\right]^{0.5}\left[\sum_{i=1}^{N}(P_i - \overline{P})^2\right]^{0.5}} \right\}^2 \qquad \textbf{6-9}$$

where P denotes the model simulated value, O denotes observed data, and the over bar denotes the mean for the entire time period of the evaluation. $i = 1,2,\ldots N$, where N is the total number of simulated and observed data pairs. R^2 is the square of the Pearson's product-moment correlation coefficient and describes the proportion of the total variance in the observed data that can be explained by the model (Legates and McCabe, 1999). R^2 is an indicator of strength of relationship between the observed and simulated values. The value of R^2 ranges between 0 and 1. If R^2 values are equal to one, the model prediction is considered to be "perfect".

6.4 Results and discussion

For each test case, the SVM and ANN models were run 20 trials to obtain the average evaluation coefficients (R^2 and $RMSE$) to assess the performance. As the R^2 and $RMSE$ are highly correlated to each other, and R^2 is popular in previous studies (e.g., Morshed and Kaluarachchi, 1998; Virginia *et al*, 2000) to evaluate the approximation ability of learning machines, the analysis of the performance of learning machines was mainly based on R^2 in order to save space. Figure 6-3 shows the R^2 values obtained by SVM and ANN for different parameter dimensions, training sample sizes, and cross validation schemes in LREW and MCEW. the R^2 values range from 0.834 to 0.996 in LREW and from 0.728 to 0.941. In the previous studies that have reported successful applications of learning machines as surrogates of complex models, the ranges of R^2 were between 0.927 and 0.988 (Virginia et al., 2000), and between 0.867 and 0.998 (Morshed and Kaluarachchi, 1998). The R^2 values obtained in this study are close to those reported in previous studies. From Figure 6-3, it is important to note that the performance of the learning machines is substantially impacted by the parameter dimensions, training sample sizes, and cross validation schemes. Based on the

simulation results, the following analysis and discussion will focus on four aspects: 1) comparing the performances between the ANNs and SVM, 2) the effect of training sample size on the performance of learning machines, 3) the effect of parameter dimension on the performance of learning machines, 4) the effect of different cross-validation schemes on the performance of learning machines. In Figure 6-3, Figure 6-4, and Figure 6-5, "6p" denotes six parameters scenario, "9p" denotes nine parameters scenario, "12p" denotes twelve parameters scenario, and "16p" denotes sixteen parameters scenario; On the horizontal axis, "1" represents 200 random samples, "2" represents 400 random samples, "3" represents 600 samples, "4" represents 800 samples, and "5" represents 1000 samples.

6.4.1 Comparison between the performances of ANN and SVM in the LREW and MCEW

From Figure 6-3, it was found that the overall performances of SVM and ANNs were very close to each other, but SVM performed slightly better than ANNs. For most cases, SVM can find higher R^2 and lower *RMSE* than ANN. In the LREW, among the total 120 test cases, SVM performed better than or equal to ANN for 109 test cases. If the no cross-validation scheme was removed, then SVM performed no less than ANN for 97 test cases from the total 100 test cases. In the MCEW, SVM performed better than or equal to ANN for 75 test cases among the total 120 test cases. If the no cross validation scheme was removed, then SVM performed no less than ANN for 70 test cases from the total 100 test cases. For each combination of parameter dimension and training sample sizes, the best solutions found by SVM and ANN with different cross validation schemes were plotted in Figure 6-4. In the LREW, the best solutions found by SVM are no less than the best solutions found by ANNs. For example, for the six parameters scenario, with 200 training samples, SVM found solutions with R^2 of 0.988 and *RMSE* of 0.08, while ANNs found solutions with R^2 of 0.987 and *RMSE* of 0.083. Similar results were also obtained for 400, 600, 800 and 1000 training samples of the six parameter scenario. For the nine, twelve and sixteen parameters scenarios in the LREW, SVM can also provide superior evaluation coefficients to ANN. In the MCEW, the SVM did not

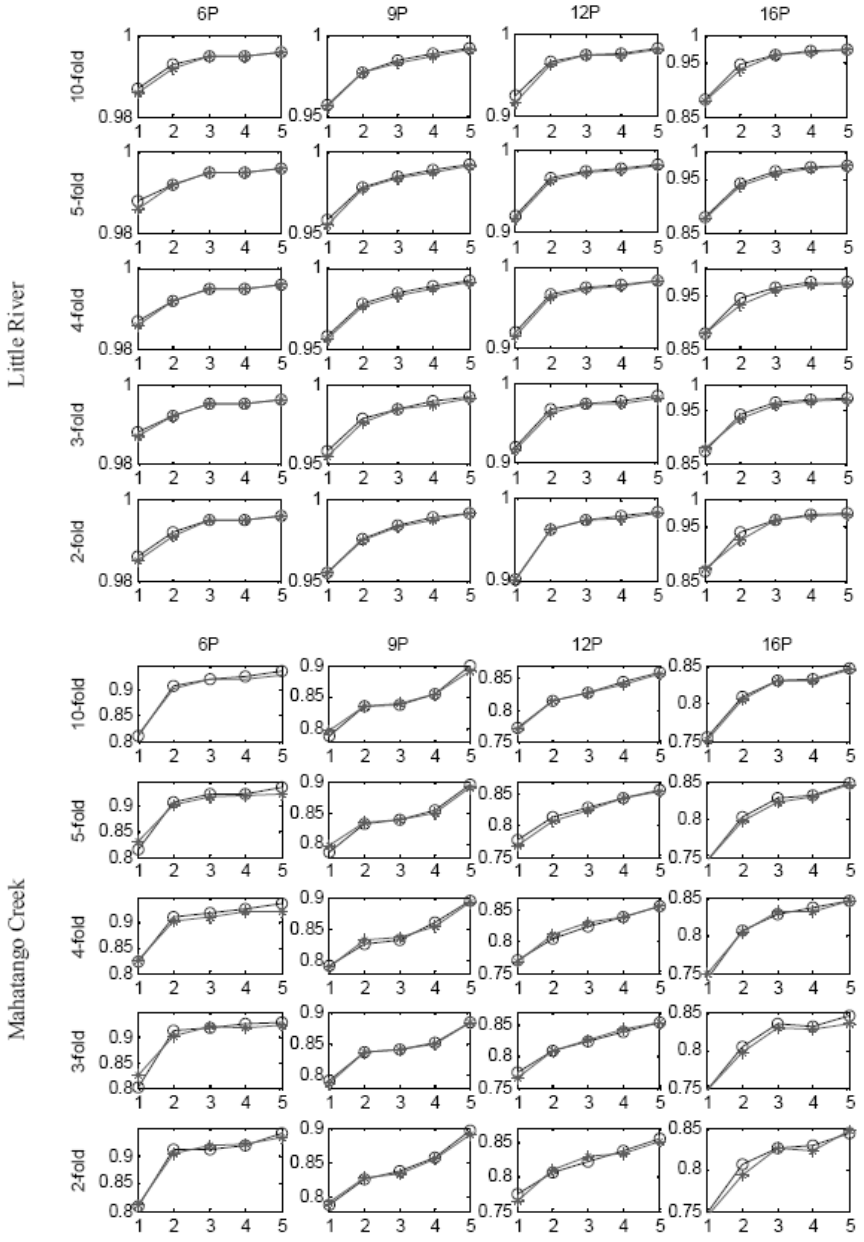

Figure 6-3. R square values obtained by SVM and ANN for different parameter dimensions, training sample sizes, and cross-validation schemes in the LREW and MCEW.

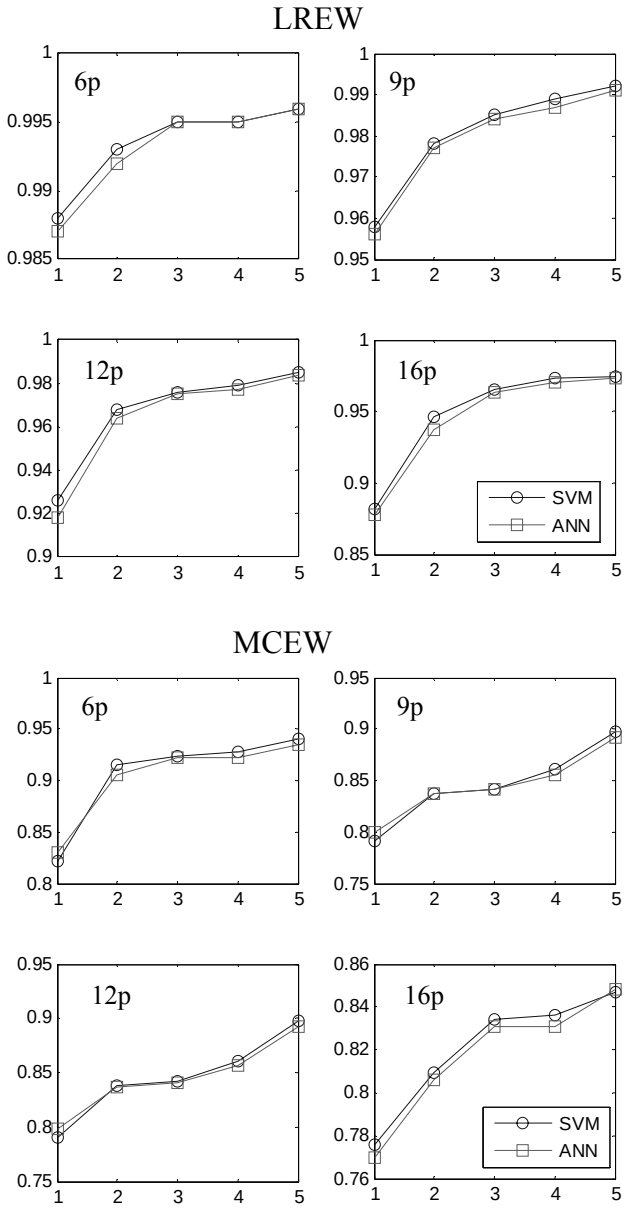

Figure 6-4. Best R square values obtained by SVM and ANN for different parameter dimensions, and training sample sizes in the LREW and MCEW.

always perform better than ANN in terms of finding best solutions. For the six and nine parameter scenarios, with 200 training samples, ANN obtained better results than SVM. For example, the best solution (with R^2 of 0.83) found by ANN is better than the best

solution (with R^2 of 0.822) found by SVM under the six parameter scenarios. SVM still outperformed ANN for 400, 600, 800 and 1000 training samples under the six and nine parameters scenarios. For the twelve and sixteen parameter scenarios, the SVM also found better solutions than ANN, given all training sample sizes. The simulation results show that SVM can obtain better evaluation coefficients for most test cases, especially when the cross validation schemes were carefully selected. Overall, SVM is preferred to ANNs for approximating the model behaviors of SWAT.

6.4.2 Effect of different cross-validation schemes on the performance of SVM

Figure 6-5 shows the performance of SVM with different cross-validation schemes for different combinations of parameter dimensions and training sample sizes. The cross-validation schemes can influence the performance of SVM. In general, the 10-fold, 5-fold, 3-fold, and 2-fold cross-validation schemes performed better than the training scheme of No cross-validation. For all of the test cases, the No cross-validation scheme exhibited lower R^2 and larger *RMSE* values than the other five training schemes with cross-validation. For example, for the test case of 12 parameters and 200 training samples, the No cross-validation scheme found solution with R^2 of 0.836 and *RMSE* of 0.273, while the least solution found by other cross-validation schemes is with R^2 of 0.903 and *RMSE* of 0.205. This indicates that the generalization ability of the learning machines can be substantially improved through applying cross-validation. The No cross-validation training scheme should not be adopted in the future. Among the five cross-validation schemes, as to which scheme should be chosen, two factors were considered in this study: 1) the prediction accuracy of each cross-validation scheme, and 2) the time consumed by each scheme, if the performances of different schemes are very similar to each other. The performances of different cross-validation schemes are impacted by the watershed characteristics, parameter dimensions, and sample sizes. In order to give a comprehensive evaluation of different cross-validation schemes, the relative performance ranks of the five cross-validation schemes were summed over all combinations of parameter dimensions and sample sizes for each watershed (Table 6-2). In general, the cross-validation schemes with more folds tend to perform better. It is

shown that the 5-fold cross-validation scheme performed best in LREW, and the 10-fold cross-validation scheme performed best in MCEW. The 10-fold cross-validation scheme doesn't always outperform other cross-validation schemes. This may be because of the stochastically training of SVM with limited number of iterations. Although the overall performance of the 10-fold cross-validation scheme is better than other cross-validation scheme, it is also found that the average R^2 values obtained by the five cross-validation schemes are close to each other (Figure 6-5). In order to test whether the mean R^2 values obtained by the five cross-validation schemes are statistically different from each other, the Analysis of Variance (ANOVA) method was implemented. The null hypothesis is that the mean R^2 values obtained by the different cross-validation schemes are the same. If all of the five cross-validation schemes were compared, the results (for most test cases the p-value is less than 0.05) would show that the null hypothesis is false for most cases. If four cross-validation schemes (3-fold, 4-fold, 5-fold, and 10-fold) were compared, then the results (the p-values for most test cases are larger than 0.05) would show that the null hypothesis is true for most test cases. Based the statistical tests, it is assumed that the 3-fold cross-validation scheme can perform as well as the higher folds cross-validation schemes. Considering the time consumed to train the SVM, the 3-fold cross-validation scheme was suggested to be used to approximate the response of SWAT model to parameter calibration.

Table 6-2. Cumulative performance rank of different cross-validation schemes in LREW and MCEW.

Watershed	10-fold	5-fold	4-fold	3-fold	2-fold
LREW	45	39	43	48	78
MCEW	44	57	60	63	70

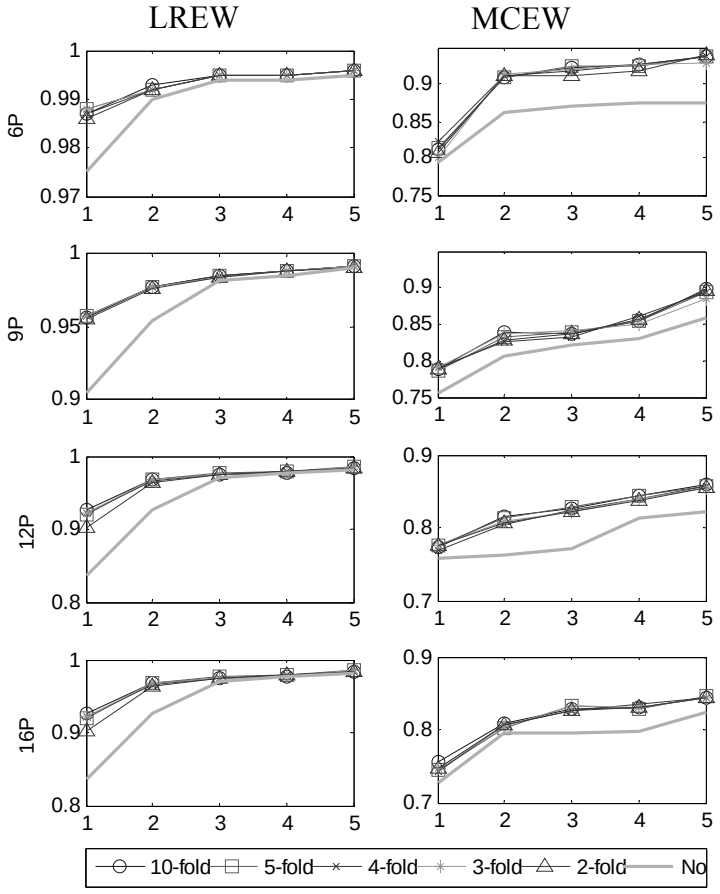

Figure 6-5. Comparison between R square values obtained by SVM with different cross validation schemes for different combinations of parameter dimension and training sample size.

6.4.3 Effect of training sample size and parameter dimension on the performance of learning machines

From Figure 6-4 and Figure 6-5, it has been shown that the training sample sizes and parameter dimensions can substantially influence the performance of learning machines to approximate the behavior of SWAT model. In general, the performance of SVM increases with the increasing of training sample size and decreases with the increasing of parameter dimension. The high parameter dimensionality makes the response surface of SWAT model to parameters more complex, which in turn make it difficult for SVM to approach. For example, in the LREW, given 200 training samples, the R^2 values are

0.988 under the six parameters scenario and 0.882 under the sixteen parameters scenario, respectively. In order to effectively apply SVM to approach the behavior of SWAT model, reducing the dimension of the parameters is important. Determining the parameter values from field data as much as possible and fixing the spatial patterns of parameters are effective ways of reducing the dimensionality of parameters. Also, the sensitivity analysis (SA) can serve as useful to screen out the insensitive parameters.

Large numbers of training samples provide more information about the response surface of SWAT model to parameters, which make SVM provide more accurate prediction. For instance, in the MCEW, under the six parameters scenario, the R^2 values are 0.909 given 400 training samples and 0.814 given 200 training samples, respectively. The effect of training sample size on the performance of SVM is influenced by the watershed characteristics, parameter dimensionality, and number of model evaluations considered in this study. In LREW, sample number larger than 600 did not provide significant gains in LREW, while there was no clear indication on how many training samples are adequate for applying SVM to approximate SWAT. In the future, further research needs to be conducted in order to determine appropriate number of training samples.

6.4.4 Simple illustration of the applicability of SVM for parameter estimation of SWAT model

In the previous sections, we have shown that SVM can obtain high R^2 values in terms of approximating the response of SWAT model to parameters. In this section, simple illustration examples were used to show the potential usefulness of incorporating SVM into the parameter calibration and uncertainty analysis process for the SWAT model. In this study, we use the Generalized Likelihood Uncertainty Estimation (GLUE) (Beven and Binley, 1992), which is a simple and flexible method that has been referred to in a vast number of literatures for parameter uncertainty analysis, to illustrate the application of SVM for parameter uncertainty analysis of SWAT. The essence of GLUE is to identify a set of acceptable or behavioral parameter sets. And, the multiple behavioral parameter sets will be run to determine the prediction limits of variables of

interest. A common method adopted by GLUE to generate candidate parameter sets is uniform random sample of the parameter space. The generated parameter sets with high objective function values (e.g. the best 10%) (Zak and Beven, 1999) will be taken as behavioral, and those non-behavioral parameter sets will be discarded. The combination of SVM with the Monte Carlo sampling method can substantially improve the efficiency of the GLUE method. As GLUE uses only those parameter sets with high objective function values, we can train SVM using a relatively small number of samples obtained with the SWAT model, and then use SVM as a surrogate to evaluate whether other parameter sets are behavioral or not. Only those parameter sets that are taken as promising to be behavioral by SVM will be evaluated by SWAT. The illustration of the application of SVM within the GLUE framework is straightforward, using two examples. For the six parameters scenario in LREW, the SVM with 3-fold cross-validation was trained using 200 samples, and this trained SVM was applied to simulate the objective function values of 10000 test samples. 730 of 1000 top 10% test parameter sets are within the simulated top 10% parameter sets, and all of these 1000 top 10% test parameter sets are included in the simulated top 26% parameter sets. For the six parameters scenario in MCEW, similar analysis was conducted. It was found that 550 of the 1000 top 10% test parameter sets are within the simulated top 10% parameter sets, and all of the top 10% test parameter sets are within the simulated top 55% parameter sets. Through the application of SVM, we can obtain more behavioral parameter sets, given limited evaluations of computationally intensive models. This advantage of incorporating SVM with SWAT model can save huge amounts of time consumed by running SWAT for parameter uncertainty analysis.

Another example is to combine the SVM with PSO to improve the efficiency of the PSO algorithm for parameter optimization of SWAT. The flowchart of incorporating the SVM into PSO algorithm is shown in Figure 6-7. The SVM, which is fitted to the past samples generated by PSO, is used to evaluate whether a newly generated parameter set needs to be evaluated by the SWAT model or not. If SVM simulated the fitness value of the new particle is larger than the 80% of the personal best fitness of that particle, then

SWAT model will be triggered. Otherwise, the personal best particle will not be updated, and the implementation of SWAT is avoided. The purpose of using SVM as a surrogate of SWAT is to save the time consumed by running SWAT model. The original PSO and PSO-SVM algorithms were implemented to optimize SWAT model of the MCEW with the six parameters scenario. Given 300 model evaluations of SWAT model, the PSO-SVM algorithm obtained the average E_{ns} value of 0.644, while the PSO algorithm found the average E_{ns} value of 0.627 (Figure 6-6). Using the ANOVA analysis, the average E_{ns} values obtained by PSO-SVM and PSO algorithms are statistically different (with a p-value of 0.042). With 1000 model evaluations, the PSO algorithm obtained an average E_{ns} value of 0.645, which is not statistically different from that obtained by the PSO-SVM algorithm with 300 model evaluations. This example shows that the efficiency of the original PSO algorithm can be substantially improved through incorporating the function approximation. It is important to note that this is one simple example of combining SVM with one type of evolutionary algorithm. Combining SVM with other evolutionary algorithms such as Genetic Algorithms (GA), Shuffled Complex Evolution (SCE) and Differential Evolution (DE) need to be performed and compared for multiple scenarios in further research.

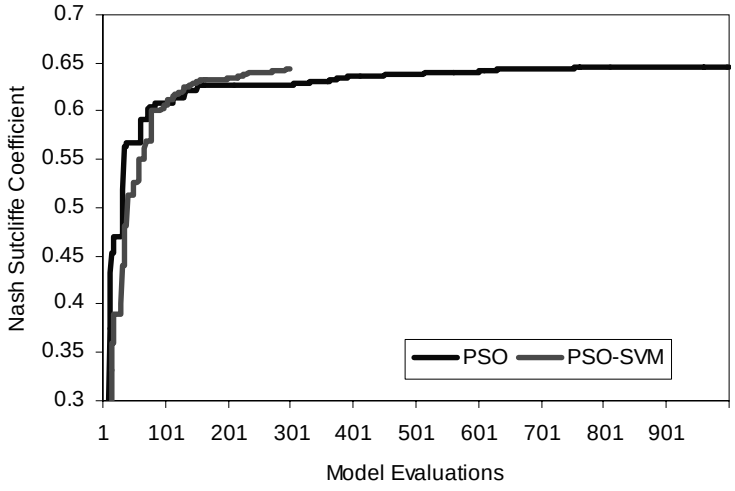

Figure 6-6. Performance of PSO and PSO-SVM against model evaluations.

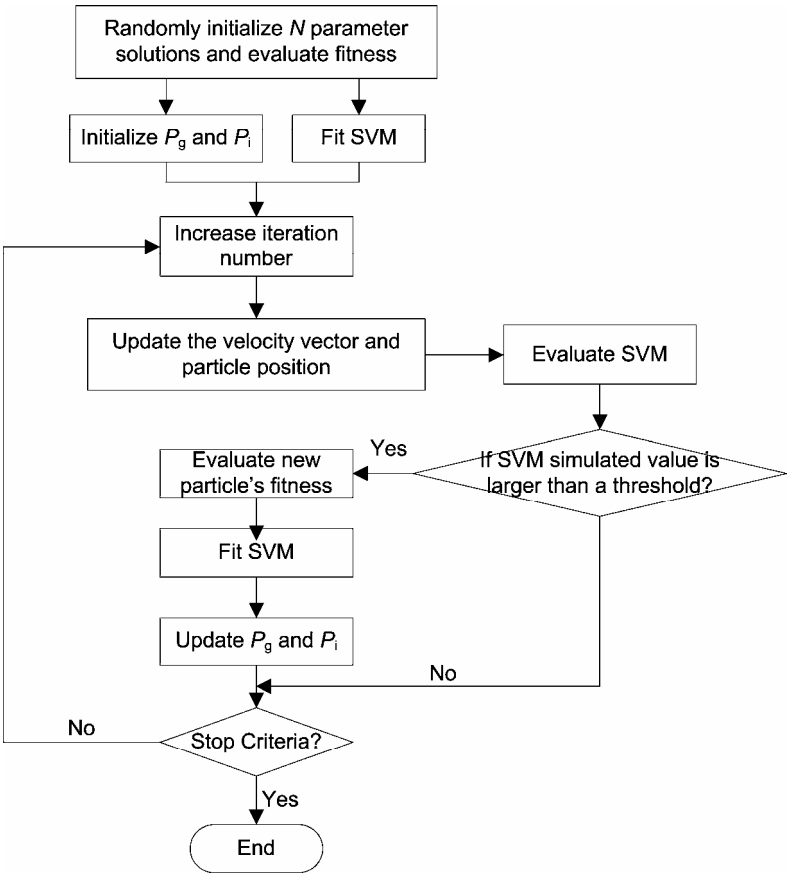

Figure 6-7. Flow chart of PSO-SVM algorithm.

6.5 Summary

In this study, two learning machines ANN and SVM were evaluated and compared for approximating the SWAT model. The PSO algorithm was applied to estimate the parameters of ANN and SVM. The results showed that both SVM and ANN can obtain high evaluation coefficients for approximating SWAT model, however, SVM exhibited better generalization ability than ANN. It is suggested that SVM be applied to approximate SWAT model.

In order to effectively and efficiently apply learning machines to approximate SWAT, the effect of parameter dimensions, training sample size, and cross-validation schemes on the performance of learning machines was investigated. Six types of cross-validation schemes (10-fold, 5-fold, 4-fold, 3-fold, 2-fold and No cross-validation) were used to

train the SVM model. Considering the prediction accuracy and time consumed for training SVM, the 3-fold cross-validation scheme, which could provide, statistically, equally good performance as the higher folds of cross-validation. Parameter dimensions can substantially impact the performance of SVM. In general, the performance of SVM decreases with the increasing of parameter dimensions. Reducing the parameter dimension through determining the parameter values from field data and the sensitivity analysis is important to improve the prediction ability of SVM. The number of training samples is another important factor influencing the performance of SVM. Large numbers of training samples can provide more accurate prediction at the cost of running the computationally intensive model many times. But, based on the results obtained in this study, it is difficult to draw conclusions on how many training samples are adequate for applying SVM to approximate SWAT.

The potential applicability of SVM model for improving the efficiency of parameter calibration and uncertainty analysis of SWAT was illustrated using simple examples. The first example showed that the parameter uncertainty analysis of SWAT using GLUE can save about 70% and 45% evaluations of SWAT in LREW and MCEW, respectively. In the second example, a new PSO-SVM algorithm was used to optimize SWAT. The results showed that the objective function values obtained by PSO-SVM with 300 evaluations of SWAT are close to that obtained by PSO with 1000 evaluations of SWAT. Overall, the results obtained in this study show that the learning machines have the ability to provide good approximation of the computationally intensive SWAT model, and hence serve as a valuable means for saving efforts in parameter calibration and uncertainty of SWAT. In the future, further research on evaluating the applicability of SVM for approximating SWAT in other watersheds and combing SVM with other parameter uncertainty analysis algorithms and evolutionary optimization algorithms needs to be conducted.

CHAPTER VII

PARAMETER UNCERTAINTY ANALYSIS OF SWAT

7.1 Introduction

Due to the errors in model structure, parameter and forcing data, the predictions of hydrologic models are affected by uncertainty. Uncertainty analysis currently enjoys substantial attention in hydrology modeling (Beven, 2006). It is widely recognized that proper consideration of uncertainty in hydrologic predictions is essential for purposes of both research and operational modeling (Wagener and Gupta, 2005). The reasonable estimates of the predictive uncertainty of a hydrologic prediction is valuable to water resources and other relevant decision making processes (Liu and Gupta, 2007). Usually, water management projects are planned and designed using scenarios that fall at the conservative end of the range of plausible outcomes. Over estimation of this uncertainty can result in over design of mitigation measures, while under estimation of this uncertainty can lead to inadequate preparation of possible conditions. Many uncertainty analysis methods that have been introduced in hydrologic modeling, which include Generalized Likelihood Uncertainty Estimation (GLUE) (Beven and Binley, 1992), Importance sampling (Kuczera and Parent, 1998), Markov Chain Monte Carlo (MCMC) (Vrugt et al, 2003b, Kuczera and Parent, 1998), Sequential Uncertainty Fitting SUFI-2 (Abbaspour et al., 2004), Parameter solutions (ParaSol) (van Griensven and Meixner, 2004), Ensemble Kalman Filter (EnKF) (Vrugt et al., 2005), Particle Filter (PF) (Moradkhani et al., 2005), Bayesian Recursive Estimation (BaRE) (Thiemann et al., 2001), and Bayesian Model Averaging (BMA) (Vrugt and Robinson, 2007; Duan et al., 2007; Ajami et al., 2007).

A popular method in the uncertainty analysis methods is to structure the hydrologic model as a probability model, then the confidence interval of model output can be

computed probabilistically (Montanari et al., 1997). The uncertainty analysis methods are differing in philosophy, assumptions and sampling strategies, and the understanding and quantification of different uncertainty sources can influence the estimation of the predictive uncertainty of hydrologic modeling (e.g. Beven and Binley, 1992; Beven and Freer, 2001; Wagener and Gupta, 2005; Beven, 2006; Vrugt and Robinson, 2007; Kavetski et al., 2006; Ajami et al., 2007). Among the many uncertainty analysis methods that have been introduced in hydrologic modeling, some of them (e.g., GLUE and SUFI-2) are using flexible likelihood function (any measure of goodness-of fit between the observed and simulated variable of interest) to assign different levels of confidence (weighting) to different parameter sets or models, reflecting their ability to acceptably reproduce "non-error-free" observations from the environmental system (Beven and Binley, 1992, Beven, 2006), while some of them (e.g., MCMC and BMA) try to use more statistically rigorous formulation to represent the probability of different parameter sets or models. GLUE methodology (Beven and Binley, 1992) and Bayesian Markov Chain Monte Carlo or MCMC (e.g. Kuczera and Parent, 1998, Vrugt et al., 2003b) are two methods that have gained much attention from hydrologic modelers for parameter uncertainty analysis of hydrologic and environmental model (Tolson and Shoemaker, 2008). These two methods usually require thousands of model evaluations to generate meaningful uncertainty estimates. When they are applied to computationally intensive hydrologic models like SWAT, the time consumed by implementation will be a major concern.

In this study, the objective was to evaluate and extend the GLUE and MCMC uncertainty analysis methods to assess the parameter uncertainty of SWAT. As discussed in previous chapters, implementation of the SWAT model is computationally intensive. Therefore, the research is focusing on extending the GLUE and MCMC method using the function approach for SWAT. It is expected that the function approximation approach can improve the efficiency of the GLUE and MCMC methods.

7.2 Description of GLUE and MCMC methods

7.2.1 GLUE

The GLUE method uses pseudo-likelihood function, and avoids the difficult task of defining the formal statistical likelihood function which is usually very difficult for real world hydrologic modeling problem. This simplification makes the GLUE method very popular for uncertainty analysis of hydrologic modeling. The basic idea of GLUE is that there are always different models that can equally mimic the hydrologic system, and can be equally acceptable or behavioral. Such equally acceptable or behavioral models are therefore called equifinal. GLUE requires that modelers subjectively define a 'likelihood' function that monotonically increases as agreement between model predictions and measured calibration data increases (Beven and Binley, 1992). The GLUE likelihood function can be, but is not required to be and is not typically, a statistically based likelihood function. A large number of GLUE studies utilize the Nash-Suttcliffe coefficient (1970) or some transformation of it to define the likelihood function (Tolson and Shoemaker, 2008). In this study, the Nash-Suttcliffe coefficient was taken as the 'pseudo-likelihood' function and used to select behavioral models. As the model structure, input forcing data, and other measured data are fixed, and little information is available for these sources of uncertainty, the uncertainty analysis of hydrologic modeling using SWAT model is focusing on parameter uncertainty. The purpose of applying GLUE is to identify multiple acceptable or 'behavioral' models that are defined by different parameter sets. A Monte Carlo experiment is conducted to independently sample model parameter space and identify various behavioral parameter sets. The modeler must subjectively determine whether a sufficient number of behavioral parameter sets are sampled. The generated parameter sets with high objective function values (e.g. the best 10%) (Zak and Beven, 1999) will be taken as behavioral, and those non-behavioral parameter sets will be discarded. The uncertainty bounds are calculated using the following procedures (Freer et al., 1996). First, the behavioral parameter sets are used to provide simulated streamflow $Q_{sim}{}^{i}(t)$, where t is the time, and $i = 1, \ldots n$, is the index of the behavioral parameter sets. Second, all the likelihoods of the behavioral

parameter sets are rescaled to have a cumulative sum of 1.0. Third, the $Q_{sim}{}^{i}(t)$ are ranked in ascending order for each time step, and a distribution function of the simulated $Q_{sim}{}^{i}(t)$ is constructed using rescaled likelihoods. Finally, uncertainty bounds corresponding to an preferred confidence level can derived.

7.2.2 MCMC

7.2.2.1 Bayesian analysis framework

The key principle of Bayesian approach is to construct the posterior probability distribution of parameter set θ given the observed input data ($\mathbf{x}_t$) and target data sets ($\mathbf{y}_t$). In the Bayesian analysis framework, the observed data and prior knowledge of parameters were applied to derive the posterior distribution of models parameter set θ for inference. Given the observed data sets $D = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$, the posterior distribution of the parameter set θ is defined as:

$$p(\theta \mid D) = \frac{p(D \mid \theta)\pi(\theta)}{\int p(D \mid \theta)\pi(\theta)d(\theta)} \qquad \textbf{7- 1}$$

where $p(\theta \mid D)$ is the posterior probability distribution of θ given observed data D, $\pi(\theta)$ is the prior probability distribution of θ, $\int p(D \mid \theta)\pi(\theta)d(\theta)$ is the normalizing constant, and $p(D \mid \theta)$ is the likelihood function of θ, which is denoted as $L(\theta)$ in the following. Through integrating the predictions of the model with respect to the posterior distribution of the parameter set θ, The posterior predictive distribution of output $\mathbf{y}_{new}$ for the new input $\mathbf{x}_{new}$, is (Lampinen and Vehtari, 2001),

$$p(\mathbf{y}_{new} \mid \mathbf{x}_{new}, D) = \int p(\mathbf{y}_{new} \mid \mathbf{x}_{new}, \theta)\, p(\theta \mid D)d(\theta) \qquad \textbf{7- 2}$$

The expectation of the posterior prediction distribution in equation (2) is

$$\hat{\mathbf{y}}_{new} = E(\mathbf{y}_{new} \mid \mathbf{x}_{new}, D) = \int f(\mathbf{x}_{new}, \theta)\, p(\theta \mid D)d(\theta) \qquad \textbf{7- 3}$$

where $f(\mathbf{x}_{new}, \theta)$ denotes the simulated target output give input data $\mathbf{x}_{new}$ and parameter set θ.

One major challenge in the Bayesian analysis is evaluating integrals for posterior distribution of the variable of interest. Usually, the posterior distribution of parameters is very complex and multimodal. In this case, the MCMC methods are usually used as tools for sampling the posterior probability of model parameters. In MCMC, the complex integrals in the marginalization are approximated via drawing samples from the probability distribution of θ, and $\hat{\mathbf{y}}_{new}$ can be approximated using a sample of the θ drawn from the posterior probability distribution of θ (Lampinen and Vehtari, 2001),

$$\hat{\mathbf{y}}_{new} = \frac{1}{K}\sum_{i=1}^{K} f(\mathbf{x}_{new,}\theta_i) \qquad \textbf{7- 4}$$

where, K denotes the number of all parameter set θ under consideration.

7.2.2.2 Evolutionary Monte Carlo

Many MCMC methods have been proposed, such as hybrid Monte Carlo (e.g., Neal, 1995, 1996), reversible jump MCMC (e.g., Green, 1995, Muller and Rios Insua, 1998; Holmes and Mallick, 1998; Andrieu, 2001), sequential Monte Carlo (e.g., de Freitas et al., 2001; Higdon et al., 2002), and evolutionary Monte Carlo (e.g., Liang and Wong, 2001a, 2001b, 2001c; Liang, 2005b). Several other MCMC based algorithms have been successfully applied to generating variables observing some complex distributions in water resources modeling. For example, the Shuffled Complex Evolution Metropolis algorithm (SCEM) (Vrugt et al., 2003b) and adaptive Metropolis samplers (Haario et al., 2001; Kingston et al, 2005; Marshall et al., 2004; Renard et al., 2006) have been successfully applied in hydrologic modeling. In this study, the Evolutionary Monte Carlo (EMC) algorithm was applied to train the BNNs. Because the EMC algorithm has been compared with several other famous MCMC methods, including the Gibbs (Chib, 1995), reversible jump MCMC (Green, 1995), parallel tempering (Geyer, 1991), and was shown to be a promising MCMC method (Liang and Wong, 2001c).

The EMC is a population based method, which was developed based on the combination of three popular algorithms: parallel tempering, reversible jump MCMC, and the genetic algorithm (Holland 1975; Goldberg, 1989). EMC generates new samples using the basic mutation, crossover operators in genetic algorithm. The acceptance of

new samples is guided by the Metropolis-Hastings rule (Metropolis et al., 1953; Hastings, 1970). In addition, EMC allows the position exchange between the candidates within the population. In the following sections, the basic EMC algorithm is introduced briefly. It is assumed that the researchers are interested in sampling from the distribution $f(\xi) \propto \exp(-H(\xi))$, where $H(\cdot)$ is called the energy function of ξ which corresponds to the negative log-posterior of a posterior distribution. Here ξ is referred to as an individual in the population, and represent the joint of one model structure and a set of parameters. The EMC is running multiple chains of different density distributions conditioned on the temperatures. A population of distributions $f_1(\xi^1),\ldots, f_N(\xi^N)$ are constructed as: $f_i(\xi^i) \propto \exp(-H(\xi^i)/t_i)$, $i = 1,\ldots,N$, where t_i is called the temperature of $f_i(\cdot)$, N is called the population size. $\mathbf{t} = (t_1,\ldots t_N)$ is a series of temperatures defined by the users with $(t_1 > \cdots > t_N)$. ξ^i denotes a sample from $f_i(\cdot)$. After randomly initializing a population of samples, the specific mutation, crossover, and exchange operators are implemented to update the position of samples. Although the mutation and crossover operators are similar to those applied in genetic algorithm, they have been modified such that they are reversible and usable as proposal functions for Metropolis-Hastings algorithm (Liang and Wong, 2001a, 2001b, 2001c; Liang, 2005b). A simple introduction of the three operators is refereed to Appendix A. Figure 7-1 shows the schematic diagram of one iteration of the EMC algorithm. In one iteration of EMC, the individuals in the population are firstly updated using mutation operator (with probability of η) and crossover operator (with probability of 1-η), Then the exchange operator is implemented to exchange the positions of $N-1$ pairs of randomly sampled individuals (ξ^i, ξ^j). In the implementation of mutation and crossover operators, new samples are generated and model needs to be evaluated, while no new samples are yielded and model evaluation is not needed during the exchange operation. The EMC algorithm has several attracting properties for effectively and efficiently generating sample from the model space(Liang and Wong, 2001a, 2001b, 2001c; Liang, 2005b): Adopting a sequence of distributions along a temperature series can help the sampler

overcoming barriers of the landscape of energy function; The crossover operator makes EMC has the learning ability of genetic algorithm; and exchange operator accelerate the mix of individuals in different sequences without extra evaluation of the energy function

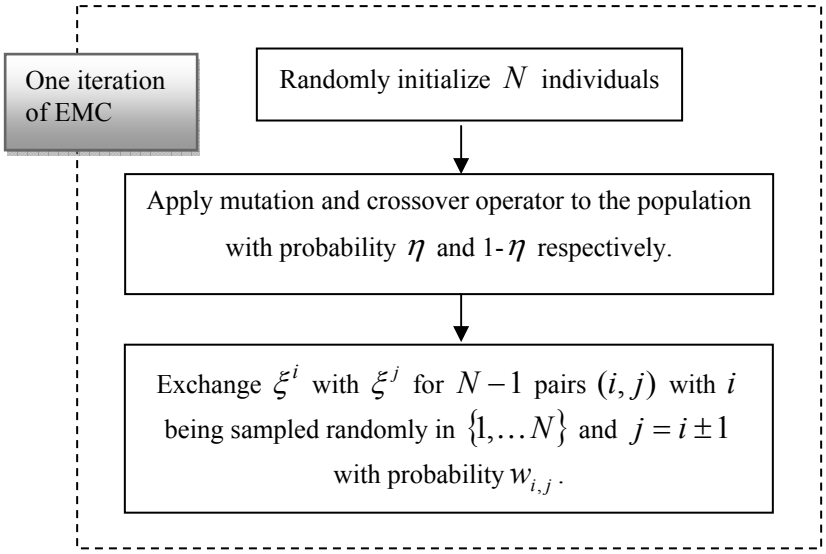

Figure 7-1. Schematic illustration of one iteration of EMC.

In order to implement the EMC algorithm, the user needs to specify several parameters that control its effectiveness and efficiency. These parameters include: the population size N, mutation rate η, Metropolis step size κ, and temperature series $\boldsymbol{t}$. Based on the recommendation in Liang and Wong (2001c) and Liang (2005b), we set $N = 20$, $\eta = 0.6$, $\kappa = 0.25$, the highest temperature $t_1 = 20$, the lowest temperature $t_N = 1$, and the intermediate temperatures are equally spaced in between t_1 and t_N. The step size was tested to make sure that the acceptance rate ranged from 0.2 to 0.4 as suggested by Gelman et al., (2004). For more detailed information on EMC algorithm and issues related to the choice of the control parameters (e.g. temperature series) please refer to Liang and Wong (2001c).

In implementing a MCMC method, it is important to check whether the sampler converge to target distribution or not. As the EMC is running with multiple chains of different density distribution conditioned on a series of temperatures, it's difficult to apply the scale reduction score to detect whether the MCMC sampler converge or not. In this study, one commonly used diagnostic of convergence through trace plots of sample MCMC values versus iteration was applied to detect the convergence of EMC sampler. It's assumed that the convergence has been reached when the trace plot flattens out (Kass et al., 1998). With the multimodal nature of neural networks' weights and structures, the convergence to the posterior weights distribution is usually very slow.

Mutation. The mutation operator is used to generate the variant of a chromosome (denoted as ξ^i, where the superscript i is the position of a chromosome in the current population). ξ^i is selected at random from the current population $\mathbf{z} = \{\xi^1, \ldots, \xi^{i-1}, \xi^i, \xi^{i+1}, \ldots, \xi^N\}$. ξ^i is modified to form a new chromosome $\xi^{i'}$ by the "Metropolis" operation defined by Metropolis et al. (1953) and Hasting (1970). The newly generated population $\mathbf{z}' = \{\xi^1, \ldots, \xi^{i-1}, \xi^{i'}, \xi^{i+1}, \ldots, \xi^N\}$ is accepted with probability,

$$\min\left\{1, \exp\left\{-\left(H(\xi^{i'}) - H(\xi^i)/t_i\right)\right\}\frac{T(\mathbf{z}\,|\,\mathbf{z}')}{T(\mathbf{z}'|\,\mathbf{z})}\right\} \qquad \textbf{7- 5}$$

where $T(\mathbf{z}\,|\,\mathbf{z}')/T(\mathbf{z}'|\,\mathbf{z})$ is the ratio of the transition probabilities. For detailed information on the "birth" and "death" operations and calculation of the three types of transition probabilities, please refer to Liang and Wong (2001a, 2001b, 2001c) and Liang (2005b).

Crossover. The crossover operator is similar to that used in the popular Genetic Algorithm. Through recombination of two chromosomes, which are randomly selected from the current population, offspring are produced. First of all, two chromosomes ξ^i and ξ^j (j is the position of a chromosome in the current population with a different value from i) are selected as parental chromosomes. Next, an integer c is drawn

randomly among $\{1,2,\ldots,M\}$, where M is the number of hidden units. The hidden unit c is called the crossover unit, and only one unit crossover operator is applied in this study. Finally, two new offspring $\xi^{i'}$ and $\xi^{j'}$ are constructed by swapping the weights connected with hidden unit c between ξ^i and ξ^j. A new population is constructed by replacing the parental chromosomes with the new offspring, and it is accepted with probability

$$\min\left\{1,\exp\left\{-(H(\xi^{i'})-H(\xi^i))/t_i-(H(\xi^{j'})-H(\xi^j))/t_j\right\}\frac{T(\mathbf{z}\,|\,\mathbf{z}')}{T(\mathbf{z}'|\,\mathbf{z})}\right\} \qquad \textbf{7- 6}$$

where $T(\mathbf{z}'|\,\mathbf{z})=P(\xi^i,\xi^j\,|\,\mathbf{z})P(\xi^{i'},\xi^{j'}\,|\,\xi^i,\xi^j)$, $P(\xi^i,\xi^j\,|\,\mathbf{z})=2/(N(N-1))$ is the selection probability of (ξ^i,ξ^j) from population $\mathbf{z}$, and $P(\xi^{i'},\xi^{j'}\,|\,\xi^i,\xi^j)$ denotes the generating probability of $(\xi^{i'},\xi^{j'})$ from the parental chromosomes (ξ^i,ξ^j). The crossover operator is symmetric, which means that $P(\xi^{i'},\xi^{j'}\,|\,\xi^i,\xi^j)$ = $P(\xi^i,\xi^j\,|\,\xi^{i'},\xi^{j'})$. $T(\mathbf{z}\,|\,\mathbf{z}')/T(\mathbf{z}'|\,\mathbf{z})=1$ for the crossover operator.

Exchange. The exchange operator is useful for exchangimg information obtained by different series of chromosomes within the population. Given the current population $\mathbf{z}$ and the attached temperature ladder $\mathbf{t}$, an exchange is made between ξ^i and ξ^j without changing the temperature t associated with the specific position within the population. The initial population and temperature ladder $(\mathbf{z},\mathbf{t})=(\xi^1,t_1,\ldots,\xi^i,t_i,\ldots,\xi^j,t_j,\ldots,\xi^N,t_N)$ are proposed to be changed to $(\mathbf{z}',\mathbf{t})=(\xi^{1'},t_1,\ldots,\xi^{i'},t_i,\ldots,\xi^{j'},t_j,\ldots,\xi^{N'},t_N)$. In this paper, the exchange is only operated on two chromosomes neighboring each other (i.e., $|i-j|=1$). The new population is accepted with probability

$$\min\left\{1,\exp\left\{(H(\xi^i)-H(\xi^j))\left(\frac{1}{t_i}-\frac{1}{t_j}\right)\right\}\frac{T(\mathbf{z}\,|\,\mathbf{z}')}{T(\mathbf{z}'|\,\mathbf{z})}\right\} \qquad \textbf{7- 7}$$

where $T(\mathbf{z}'|\mathbf{z}) = p(\xi^i)w_{i,j} + p(\xi_j)w_{j,i}$, $P(\xi^i)$ is the probability that ξ^i is chosen to exchange with the other chromosome, $w_{i,j}$ denotes the probability that ξ^j is chosen to exchange with ξ^i, $w_{i,i+1} = w_{i,i-1} = 0.5$ for $1 < i < N$ for and $w_{1,2} = w_{N,N-1} = 1$. Thus $T(\mathbf{z}|\mathbf{z}')/T(\mathbf{z}'|\mathbf{z}) = 1$ for the exchange operator.

7.3 Results and discussion

7.3.1 Application of GLUE and SVM for parameter uncertainty analysis of SWAT

To reduce the number of runs of the computationally intensive SWAT model required by the GLUE method, the SVM model was trained using a small number of samples (1000) generated by the SWAT model. 20000 parameter sets were generated using SLHD algorithm. The parameter sets which give non-behavioral evaluation coefficients are discarded before running the computationally intensive SWAT model. It is expected that the combination of GLUE and SVM (GLUE-SVM) can effectively reduce the number of actual model runs of SWAT without the loss of accuracy. The newly proposed GLUE-SVM algorithm was applied in the LREW and MCEW, and its results were compared with the original GLUE method.

In hydrologic modeling, different types of uncertainty limits can be recognized (e.g. Beven, 2006; Liu and Gupta, 2007). In this study, we are concerned with the modeling uncertainty and predictive uncertainty (Liu and Gupta, 2007). The modeling uncertainty limits, obtained through training hydrologic model to match observed streamflow data, were expected to include a specified proportion of the training data set. The predictive uncertainty limits, obtained through applying the trained models to another independent data set, were expected to contain a specified proportion of future observations. Ideally, the uncertainty interval should be consistent with observations and be as small as possible (Vrugt et al., 2007). Two coefficients were used to compare the uncertainty intervals estimated by different methods: 1) the percentage of coverage (POC) of observations in the uncertainty interval, and 2) the average width of the uncertainty interval, which is denoted as D-bar in the following sections. The POC coefficient is firstly evaluated. The uncertainty interval with a POC coefficient close to the expected

proportion is preferred. If the POC of two uncertainty intervals are very close to each other, then the uncertainty interval with narrower D-bar value is considered better.

The POC and D bar values of the uncertainty intervals estimated by GLUE with different threshold values in MCEW and LREW for the calibration and validation periods are listed in Table 7-1 toTable 7-4. It is found that the POC and D-bar values obtained by the original GLUE and GLUE-SVM methods for different threshold values are very close to each other.

The R^2 values between the POC values estimated by GLUE and GLUE-SVM are 0.87 for calibration period and 0.86 for validation period in MCEW, and 0.90 for calibration period and 0.96 for validation period in LREW, respectively. The R^2 values between the D-bar values estimated by GLUE and GLUE-SVM are 0.95 for calibration period and 0.96 for validation period in MCEW, and 0.88 for calibration period and 0.91 for validation period in LREW, respectively. The high correlation between the properties of the uncertainty intervals estimated using GLUE and GLUE-SVM show that the GLUE-SVM can serve as a promising alternative for the original GLUE method.

In MCEW, among the 20000 randomly generated parameter sets, only about 2% of them have E_{ns} values larger than or equal to 0.36, which is the criterion to evaluate whether the hydrologic simulation is satisfactory or not. All these top 2% ranked parameter sets are within the top 40% parameter sets ranked by the fitness simulated by GLUE-SVM. In LREW, among the 20000 randomly generated parameter sets, only about 3% of them have E_{ns} values larger than or equal to 0.36. All these top 3% ranked parameter sets are within the top 20% parameter sets ranked by the fitness simulated by GLUE-SVM. If our purpose is to identity behavioral parameter sets and the non-behavioral parameter sets will be discarded, then the GLUE-SVM method can save about more than 50% runs of the computational intensive SWAT model.

Table 7-1. POC and D bar values of the uncertainty intervals estimated by GLUE and GLUE-SVM with different threshold values in MCEW for the calibration period.

Threshold	POC (%)		Threshold	D-bar (cms)	
	GLUE	GLUE-SVM		GLUE	GLUE-SVM
1%	55.05	59.04	1%	0.08	0.11
2%	69.86	64.79	2%	0.11	0.13
3%	73.42	67.4	3%	0.14	0.14
4%	75.62	69.32	4%	0.15	0.143
5%	77.12	71.1	5%	0.16	0.154
10%	81.64	80	10%	0.19	0.175
20%	85.62	84.79	20%	0.21	0.21
30%	86.16	85.07	30%	0.23	0.22

Table 7-2. POC and D bar values of the uncertainty intervals estimated by GLUE and GLUE-SVM with different threshold values in MCEW for the validation period.

Threshold	POC (%)		Threshold	D bar (cms)	
	GLUE	GLUE-SVM		GLUE	GLUE-SVM
1%	47.33	53.08	1%	0.08	0.1
2%	61.97	57.59	2%	0.11	0.12
3%	65.39	59.37	3%	0.14	0.13
4%	67.17	61.01	4%	0.15	0.14
5%	69.22	61.97	5%	0.17	0.15
10%	72.28	71.18	10%	0.19	0.17
20%	82.22	80.71	20%	0.22	0.21
30%	83.17	81.4	30%	0.24	0.22

Table 7-3. POC and D bar values of the uncertainty intervals estimated by GLUE and GLUE-SVM with different threshold values in LREW for the calibration period.

Threshold	POC (%)		Threshold	D bar (cms)	
	GLUE	GLUE-SVM		GLUE	GLUE-SVM
1%	30	34	1%	3.94	3.34
2%	33	40	2%	4.27	4.01
3%	38	41.04	3%	4.6	4.05
4%	42	41.59	4%	5.28	4.13
5%	43.09	41.72	5%	5.42	4.17
10%	45.42	47.33	10%	5.85	5.5
20%	50.42	51.57	20%	6.5	6.3
30%	51.03	51.85	30%	6.6	6.43

Table 7-4. POC and D bar values of the uncertainty intervals estimated by GLUE and GLUE-SVM with different threshold values in LREW for the validation period.

Threshold	POC (%)		Threshold	D bar (cms)	
	GLUE	GLUE-SVM		GLUE	GLUE-SVM
1%	32.94	32.39	1%	3.92	2.38
2%	34.3	37.5	2%	4.15	2.83
3%	35.4	38.5	3%	4.3	2.86
4%	37.5	38.96	4%	4.57	2.91
5%	38.96	39.51	5%	4.7	2.97
10%	44.89	46.9	10%	5.05	4.65
20%	48.27	50	20%	5.51	5.27
30%	49.36	50.18	30%	5.59	5.35

7.3.2 Parameter uncertainty analysis of SWAT using EMC and SVM

7.3.2.1 Evaluation of EMC for two illustrative examples

Before running the EMC algorithm for training BNNs, two illustrative test examples were used to show the effectiveness of EMC for generating samples observing complex distributions.The first illustrative example is a mix of two multivariate normal distributions with mean vectors of $\mathbf{0} = (0,0,0,0,0)$ and $\mathbf{8} = (8,8,8,8,8)$ respectively,

$$\pi(\mathbf{x}) = 0.5 \cdot N(\mathbf{0}, \mathbf{I}_5) + 0.5 \cdot N(\mathbf{8}, \mathbf{I}_5) \qquad \textbf{7- 8}$$

where $\mathbf{x} = (x_1, x_2, x_3, x_4, x_5)$ is a five dimensional vector, $N(\mathbf{u}, \mathbf{\Sigma})$ denote the normal distribution with mean of $\mathbf{u}$ and covariance matrix $\mathbf{\Sigma}$. This example was used by Renard et al. (2006) to test the effectiveness of three MCMC samplers. The initial population was generated within $[0,1] \times [0,1]$. 30,000 iterations of EMC were implemented to estimate the statistical properties of $\mathbf{x}$. The EMC was implemented 50 times. In order to save space, only the statistics of the first dimension and fifth dimension of $\mathbf{x}$ were listed in Figure 7-2 and Table 7-5. The results show that the EMC algorithm can accurately generate samples observing this bimodal distribution.

Table 7-5. Parameters estimation of the five-dimensional bimodal distribution.

Parameter	True value	Estimate	SD
u_1	4	4.02	0.034
u_5	4	4.014	0.035
Σ_{11}	17	16.989	0.033
Σ_{55}	17	16.99	0.035
Σ_{15}	16	15.993	0.028

NOTE: Here u_1 and u_5 are the first and fifth component of the mean of $\mathbf{x}$; Σ_{11}, Σ_{55} and Σ_{15} are the variances and covariance of the first and second component of $\mathbf{x}$; SD denotes the standard deviation of the corresponding estimate

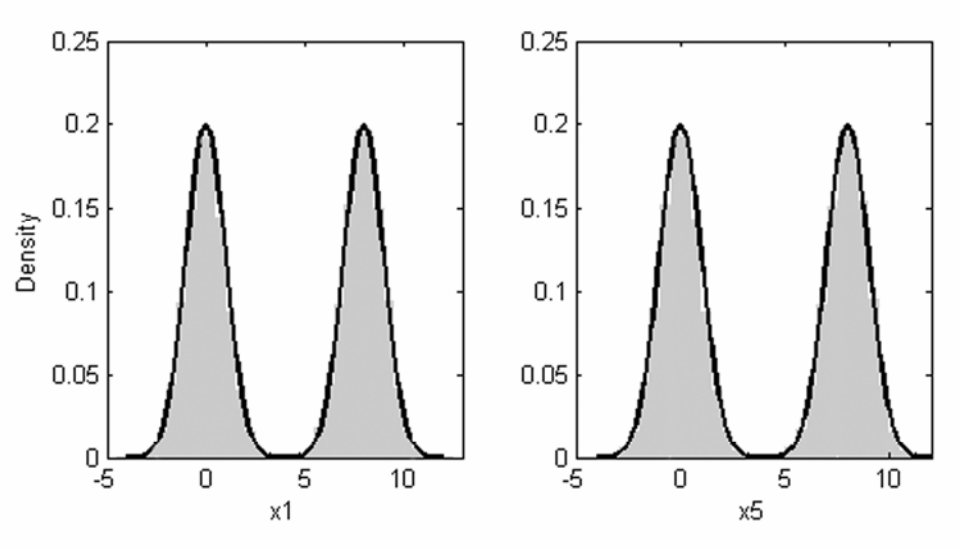

Figure 7-2. Simulated values of the first and fifth component from the two-modal distribution. The solid line is the true value, and the grey area is the density estimated by EMC.

A two dimensional multimodal mixture normal distribution was used to test the EMC algorithm to show its effectiveness for sampling candidates from distribution with complex landscape. The multimodal distribution applied here is

$$f(\mathbf{x}) = \frac{1}{\sqrt{2\pi}\sigma}\sum_{i=1}^{20}\omega_i \exp\left\{-\frac{1}{2\sigma^2}(\mathbf{x}-\mathbf{u}_i)'(\mathbf{x}-\mathbf{u}_i)\right\} \qquad \textbf{7- 9}$$

where σ=0.1, $\omega_1 = \cdots = \omega_{20}$=0.05. The mean vectors $u_1, u_2, \ldots u_{20}$ are uniformly drawn from the rectangle space $[0,10]\times[0,10]$. This multimodal distribution is similar to that used in Liang and Wong (2001c). The initial population was generated within $[0,1]\times[0,1]$. 50,000 iterations of EMC were implemented to estimate the statistical properties of $\mathbf{x}$. The EMC was implemented 50 times. Figure 7-3 shows the scatter plot of 10,000 samples, which reveals that the EMC can effectively sample all the 20 local modes. The estimate of means, variances of the two components of $\mathbf{x}$, and the standard deviation of the estimated values are shown in Table 7-6, which show that the EMC

algorithm can consistently obtain accurate estimate of the interesting statistical properties of the $\mathbf{x}$.

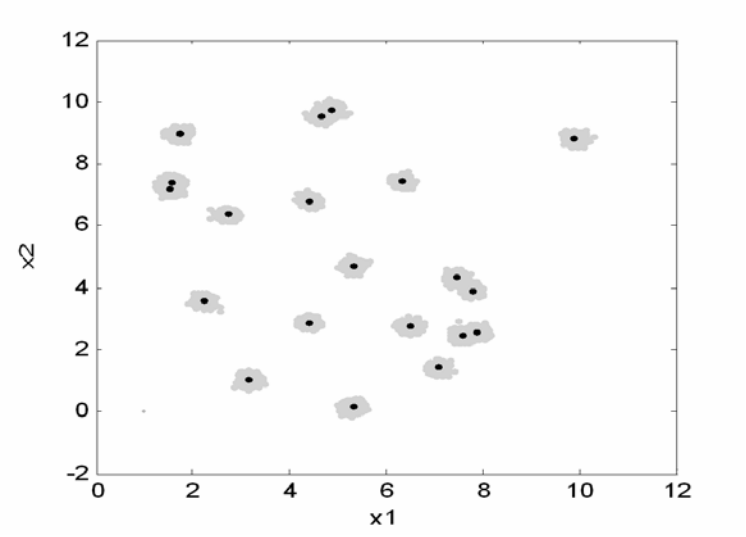

Figure 7-3. Scatter plot of the samples generated by EMC for the 20-modal distribution.

Table 7-6. Parameters estimation of the two-dimensional multimodal distribution.

Parameter	True value	Estimate	SD
u_1	5.123	5.128	0.015
u_2	5.093	5.089	0.021
Σ_{11}	5.623	5.627	0.032
Σ_{22}	8.641	8.648	0.036
Σ_{12}	-1.579	-1.583	0.025

NOTE: Here u_1 and u_2 are the first and second component of the mean of $\mathbf{x}$; Σ_{11}, Σ_{22} and Σ_{12} are the variances and covariance of the first and second component of $\mathbf{x}$; SD denotes the standard deviation of the corresponding estimate.

7.3.2.2 Application of EMC and SVM in the MCEW

In order to implement MCMC analysis of SWAT model, the posterior distribution of the parameter $p(\theta \mid D)$ needs to be defined. In this study the prior distribution $\pi(\theta)$ is

assumed to be non-informative uniform distribution. This assumption reflects the lack of prior knowledge of the distribution of parameters. And the posterior of the parameter distribution is entirely determined by the observed data. A popular method to specify the likelihood function $L(\theta)$ is to assume the model residuals (i.e. $f(\mathbf{x}_t, \theta)$ - $\mathbf{y}_t$) are normally and independently distributed with zero mean and constant variance $N(0, \sigma^2)$. Then, the likelihood of a set of $L(\theta)$ for describing the observed data **y** can be computed as:

$$L(\theta) = p(\mathbf{y} \mid \theta, \mathbf{x}) = \left\{ \prod_{t=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}(y_t - f(\mathbf{x}_t))^2\right) \right\} \times \left\{ \frac{v_2^{v_1}}{\Gamma(v_1)} (\sigma^2)^{-v_1-1} \exp\left(\frac{-v_2}{\sigma^2}\right) \right\} \quad \textbf{7- 10}$$

Then the log form of the posterior probability can be formulated as

$$\log(p(\theta)) \propto \text{Constant} - \frac{1}{2\sigma^2} \sum_{t=1}^{n} \left(y_t - f(\mathbf{x}_t, \theta, \Lambda)\right)^2 - \left(\frac{n}{2} + v_1 + 1\right) \log(\sigma^2) - \frac{v_2}{\sigma^2} \quad \textbf{7- 11}$$

where n is the number of observed target data, and σ^2 is referred to as hyperparameter, which is assumed to observe Inverse Gamma (IG) distribution ($\sigma^2 \sim IG(v_1, v_2)$), v_1 and v_2 are the shape parameter and scale parameter respectively that define the Inverse Gamma distribution. In order to effectively implement EMC, the input and output data were normalized by $q_t^{'} = (q_t - \overline{q}) / S_q$, where $\overline{q}$ and S_q denote the mean and standard deviation of the input and output data. This type of data processing tends to avoid that the hydrologic model is trained to accommodate different scales of the observed data. A vague prior on σ^2 was chosen through setting $v_1 = v_2 = 0.05$.

The EMC algorithm was implemented to train the SWAT model using observed daily streamflow from 1997 to 1998. For training the SWAT model, the EMC was run for 11,000 iterations each time. The trace of the mean log posterior density was inspected, and it was found that convergence was reached after about 6,000 iterations. The first 6,000 iterations were taken as a burn-in stage and were discarded, and the left

5,000 sets of parameters samples were collected to derive the posterior distribution of (θ). The 5,000 sets of (θ) were used to run the SWAT model and calculate the daily streamflow output. The mean of the simulations of the 5,000 streamflow values were used to estimate the 95% uncertainty interval. The computation time consumed by implementing the EMC based MCMC analysis of SWAT is enormous (several months). In order to save the computational cost, the SVM was trained using 1000 samples generated from SWAT, and then the trained SVM was used as the surrogate of SWAT to estimate the posterior distribution of parameters and estimate the uncertainty interval of simulated streamflow.

The uncertainty intervals of daily streamflow estimated by EMC and EMC-SVM algorithms for both calibration and validation periods are shown in Figure 7-4 and Figure 7-5, where shaded are denotes the 95% uncertainty interval, and solid dots dente the observed streamflow. It is found that the 95% uncertainty interval estimated by original EMC can contain about 20% observed data for both calibration and validation periods. While for the EMC-SVM algorithm, the 95% uncertainty interval can only contain less than 3% of observed data. This reveals that the applicability of SVM for MCMC analysis of SWAT is not satisfactory. Although SVM can provide fairly good approximation to the SWAT model, the discrepancy between the fitness values simulated by SVM and SWAT may lead to evident bias for inferring the parameter distribution and the corresponding uncertainties of the hydrologic modeling.

According to the discussion in Chapter I, the uncertainty of hydrologic modeling is affected by not only the parameters but also the forcing data and model structure. The application of EMC for SWAT shows that only considering parameter uncertainty is not adequate for estimate uncertainty of hydrologic modeling. The percentage of observed data that are contained in the 95% interval is much less than the expected 95%. This, to some extent, indicates the importance of taking forcing data and model structure uncertainty into consideration.

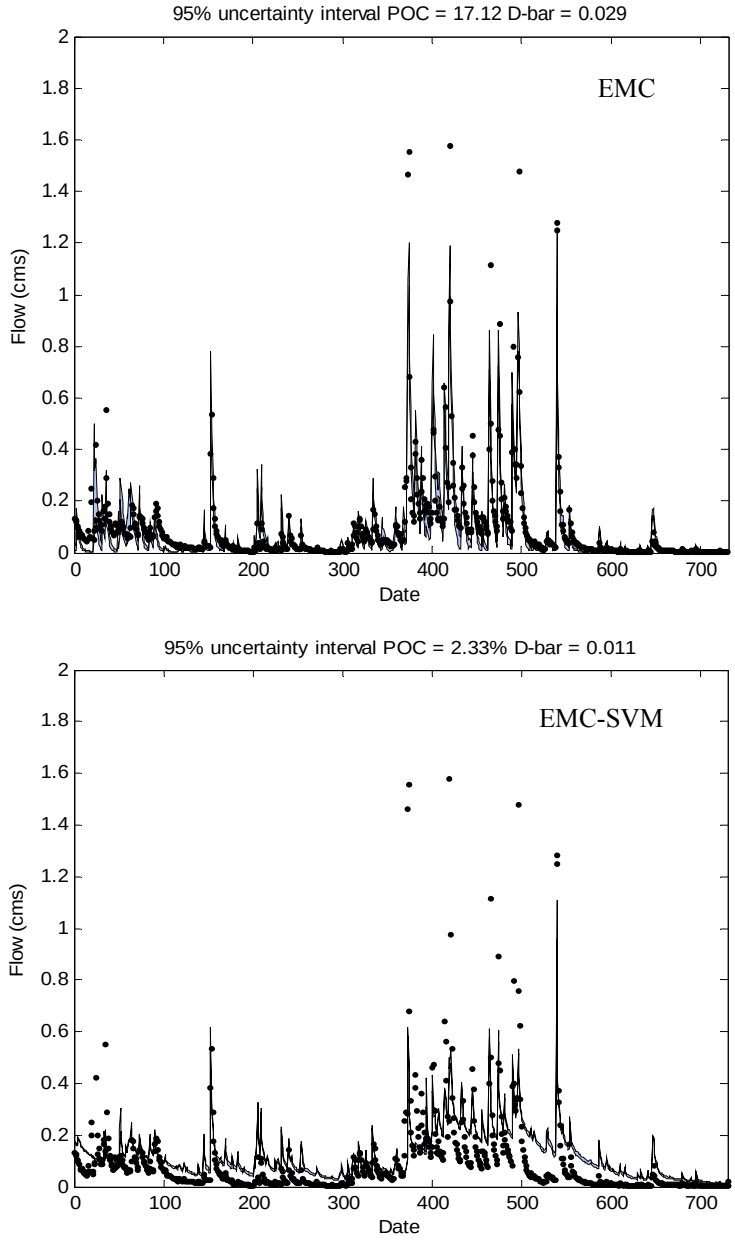

Figure 7-4. 95% uncertainty intervals estimated by EMC and EMC-SVM for the calibration period in MCEW.

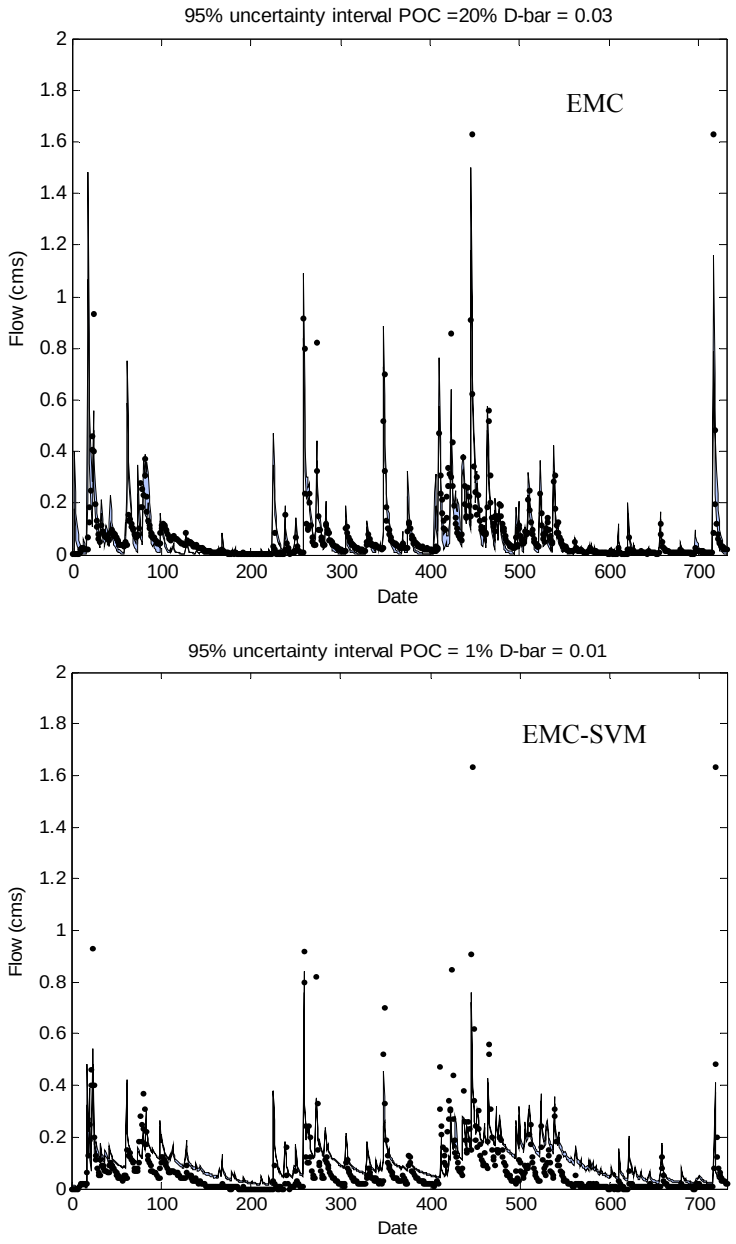

Figure 7-5. 95% uncertainty intervals estimated by EMC and EMC-SVM for the validation period in MCEW.

7.4 Summary

Parameter uncertainty analysis of hydrologic modeling of SWAT was analyzed. Among the many uncertainty analysis methods that have been introduced in hydrologic modeling, a pseudo likelihood method (GLUE) and a Bayesian MCMC based method (EMC) were applied to infer the uncertainty of parameters of SWAT, and estimate the uncertainty interval of the simulated daily streamflow. One concern of the application of GLUE and MCMC methods for parameter uncertainty analysis is the enormous time consumed by larger number of runs of the computationally intensive SWAT model. In order to save computational resources, the SVM was used as surrogate model of SWAT and combined with the GLUE and MCMC methods. The results show that the GLUE-SVM can obtain similar results to that obtained by the original GLUE method. The percentage of coverage and interval width of the uncertainty bounds estimated by the original GLUE and GLUE-SVM methods are very close to each other. The R^2 values between the POC values estimated by GLUE and GLUE-SVM are larger than 0.85, and The R^2 values between the D-bar values estimated by GLUE and GLUE-SVM are larger than 0.87. The GLUE-SVM can be taken as a promising alternative of the original GLUE method. The uncertainty intervals estimated by the EMC-SVM and the original EMC method showed evident difference from each other. The may be because the SVM is not an error free surrogate of SWAT, and the discrepancy between the simulated results of SVM and SWAT lead to the substantially different uncertainty estimation. It should be noted that that both GLUE and EMC can not accurately quantify the prediction uncertainty of SWAT model if parameter is taken as the only source of uncertainty. The results obtained in this study stress the research on taking uncertainties associated with forcing data and model structure into account.

CHAPTER VIII

MODEL STRUCTURE AND UNCERTAINTY ANALYSIS

8.1 Introduction

The uncertainty of hydrologic modeling is associated with the uncertainties from forcing data, model structure, and parameters. In the previous chapters, the parameter calibration and uncertainty analysis methods have been examined for parameter estimation of the SWAT model. The results show that not considering the uncertainties of forcing data and model structure leads to the failure to evaluate the predictive uncertainty. Understanding and evaluating the effect of uncertainty of forcing data and model structure on hydrologic modeling deserve further research. There are many input data (e.g., precipitation, temperature, land cover, soil, and elevation) for SWAT. For any of these data, a large amount of observed data needs to be collected for reasonable estimation of their uncertainties characteristics. Therefore, the major objective of this chapter is to evaluate the effect of considering model structure uncertainty on uncertainty estimation of streamflow simulation.

In this study, ANN, which has been successfully applied in a wide range of hydrologic problems (ASCE task Committee on Application of artificial Neural Networks in Hydrology, 2000a, 2000b), was selected an alternative hydrologic model to SWAT. Bayesian analysis of the neural network can yield predictive distribution of the variables of interest and make the computation of confident intervals possible (Lampinen and Vehtari, 2001). Since the Bayesian evidence framework proposed by MackKay (1992), the Bayesian Neural Network (BNN) have been widely applied in training, model selection, and prediction (e.g., Neal, 1995; Bishop, 1995; Muller and Rios Insua, 1998; Holmes and Mallick; 1998, de Freitas et al., 2000; Andrieu 2001; Liang, 2003; Liang, 2004; Liang, 2005a).

In this study, we changed the hydrologic model from SWAT to BNN to examine the difference between the uncertainty intervals obtained by SWAT and BNN. Because the implementation of EMC for SWAT model takes enormous computational time of months or even years, the comparison between SWAT and BNN was only conducted in the MCEW. In order to further understand the effect of different treatments of uncertainties related to parameters (network weights) and structures on the uncertainty estimation of streamflow simulation, different BNNs (with variable or fixed model structure, informative or non-informative prior knowledge) were applied in two experimental watersheds (LREW and RCEW) for daily streamflow simulation to derive results for analysis and discussion.

8.2 Bayesian neural network

8.2.1 Neural networks structure

Usually, the neural network structure is fixed, which means that the number of connections between the neutrons is fixed. A set of indication functions can be linked with each connection to represent the validity of a specific connection (Liang, 2004, 2005b). Then, the neural network model form presented in section 6-1 can be transformed to:

$$f(\mathbf{x}_t) = \alpha_0 I_{\alpha_0} + \sum_{i=1}^{p} x_{it}\alpha_i I_{\alpha_i} + \sum_{j}^{M} \beta_j I_{\beta_j} \psi(\gamma_{j0} I_{\gamma_{j0}} + \sum_{i=1}^{p} x_{it}\gamma_{ji} I_{\gamma_{ji}}) \qquad \textbf{8-1}$$

where I_ζ is the indicator function associated with the connection ζ. If I_ζ=1, then the connection is in effect, otherwise, I_ζ=0 and the connection is not effective. The activation function ($\psi(\cdot)$) applied in this study is the hyperbolic tangent function. The $\tanh(\cdot)$ function ensures that the output of a hidden unit is 0 if all connections to the hidden unit from input units have been eliminated. Let Λ be the vector consisting of all indicators in equation (2), then Λ specifies the structure of the network. Let $\alpha = (\alpha_0, \alpha_1, \ldots, \alpha_p)$, $\beta = (\beta_0, \beta_1, \ldots, \beta_M)$, $\gamma_j = (\gamma_{j0}, \gamma_{j1}, \ldots, \gamma_{jp})$, $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_M)$, and $\theta = (\alpha_\Lambda, \beta_\Lambda, \gamma_\Lambda, \sigma^2)$, where α_Λ, β_Λ, and γ_Λ denote the non-zero subsets of α, β, and γ, respectively. Thus the combination of (θ, Λ) completely defines equation (2). In

the following, a neural network model is defined by (θ, Λ), and $f(\mathbf{x}_t)$ is represented by $f(\mathbf{x}_t, \theta, \Lambda)$. Equation (1) is a special case of equation (2). The major difference between equation (1) and equation (2) is that equation (2) is trained by sampling from the joint posterior of the neural network structures and weights while equation (1) is trained by sampling from the posterior of the weights.

8.2.2 Bayesian training of neural networks

In the traditional deterministic training of a neural networks, a single set of optimal (θ, Λ) is identified that is most likely to reproduce the observed target data. From the Bayesian viewpoint, the training of neural networks can be taken as a problem of inference. The key principle of Bayesian approach is to construct the posterior probability distribution of (θ, Λ) given the observed input and target data sets. In the Bayesian training framework, the observed data and prior knowledge of parameters and model structure were applied to derive the posterior distribution of models (θ, Λ) for inference. Given the training data sets $D = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$, the posterior distribution of the weights and model structure (θ, Λ) is defined as:

$$p(\theta, \Lambda \mid D) = \frac{p(D \mid \theta, \Lambda)\pi(\theta, \Lambda)}{\int p(D \mid \theta, \Lambda)\pi(\theta, \Lambda)d(\theta, \Lambda)} \qquad \textbf{8- 2}$$

Where $p(\theta, \Lambda \mid D)$ is the posterior probability distribution of (θ, Λ) given observed data D, $\pi(\theta, \Lambda)$ is the prior probability distribution of (θ, Λ), $\int p(D \mid \theta, \Lambda)\pi(\theta, \Lambda)d(\theta, \Lambda)$ is the normalizing constant, and $p(D \mid \theta, \Lambda)$ is the likelihood function of (θ, Λ), which is denoted as $L(\theta, \Lambda)$ in the following. The posterior distribution of the model (θ, Λ) is estimated using EMC. The BMA estimation of target variable is

$$\hat{\mathbf{y}}_{new} = \frac{1}{K}\sum_{i=1}^{K} f(\mathbf{x}_{new,}, \theta_i, \Lambda_i) \qquad \textbf{8- 3}$$

where, K denotes the number of all models (θ, Λ) under consideration.

8.2.3 Posterior probability distribution of neural networks

To some extent, the quantification of uncertainty is dependent on understanding prior knowledge of uncertainty. In practical application of neural networks, incorporating human prior knowledge in neural networks models was suggested to improve their performance (Wang, 1995, Muler and Insua, 1998, Liang, 2005b). Usually, the large weights and bias values and complex model structures are penalized in application of neural networks. For example in Liang's study (2005b), prior distributions were specified for the weights: $\alpha_i \sim N(0,\sigma_\alpha{}^2)$ for $i=0,\ldots,p$, $\beta_j \sim N(0,\sigma_\beta{}^2)$ for $j=0,\ldots,M$, $\gamma_{ij} \sim N(0,\sigma_\gamma{}^2)$ for $j=0,\ldots,M$ and $i=0,\ldots,p$, where $\sigma_\alpha{}^2$, $\sigma_\beta{}^2$, and $\sigma_\gamma{}^2$ are hyper-parameters to be specified by users. With the prior distribution of each component of θ , the prior distribution of θ can be easily calculated under the assumption that α_i , β_j , and γ_{ij} are independent with each other. Incorporating the parameter prior knowledge into the posterior probability lead to

$$\log P(\theta,\Lambda \mid D) = \text{Constant} - (\frac{n}{2}+\nu_1+1)\log\sigma^2 - \frac{\nu_2}{\sigma^2} - \frac{1}{2\sigma^2}\sum_{t=1}^{n}(y_t - f(\ _t))^2 - \frac{1}{2}\sum_{i=0}^{p} I_{\alpha i}(\log\sigma_\alpha{}^2 + \frac{\alpha_i{}^2}{\sigma_\alpha{}^2})$$
$$-\frac{1}{2}\sum_{j=1}^{M} I_{\beta j}\Phi(\sum_{i=0}^{p} I_{\gamma_{ji}})(\log\sigma_\beta{}^2 + \frac{\beta_j{}^2}{\sigma_\beta{}^2}) - \frac{1}{2}\sum_{j=1}^{M}\sum_{i=0}^{p} I_{\beta_j} I_{\gamma_{ji}} \times (\log\sigma_\gamma{}^2 + \frac{\gamma_{ji}{}^2}{\sigma_\gamma{}^2}) - \frac{m}{2}\log(2\pi) \qquad \textbf{8- 4}$$

Where $m = \sum_{i=0}^{p} I_{\alpha_i} + \sum_{j}^{M} I_{\beta_j}\Phi(\sum_{0}^{p} I\gamma_{ji}) + \sum_{j=1}^{M}\sum_{i=0}^{p} I_{\beta_j} I_{\gamma_{ji}}$ is the total number of effective connections, $\Phi(t)$ is 1 if $t>0$ and 0 otherwise. For fixed model structure, all indicator functions are equal to 1 and m is a constant.

The prior knowledge of the structure of neural networks can also be taken into account. Based on Muler and Insua (1998) and Liang (2005b), the neural network's structure Λ is set to be subject to a prior probability that is proportional to a truncated Poisson with rate λ ,

$$p(\Lambda)=\begin{cases}\dfrac{1}{Z}\dfrac{\lambda^m}{m!}, & m=3,4,\ldots,U\\ 0, & \text{otherwise}\end{cases} \qquad \textbf{8- 5}$$

where λ is hyper-parameter, $U=(M+1)(p+1)+M$ is the number of connections of the full model in which all connections are valid, and $Z=\sum_{\Lambda\in\Omega}\lambda^m/m!$ where Ω denotes the set of all possible model structures with $m=3,4,\ldots,U$. The minimum number of m is set to be three to limit the network size. Furthermore, it is assumed that the prior distributions of θ and Λ, and the likelihood $L(\theta,\Lambda)$ are independent. Then, the posterior distribution of (θ,Λ) can be formalized by multiplying the prior distributions of θ and Λ and $L(\theta,\Lambda)$. According to (Liang, 2005b), the log form of this posterior probability can be written as:

$$\begin{aligned}\log P(\theta,\Lambda\mid D)=\text{Constant}&-(\frac{n}{2}+\nu_1+1)\log\sigma^2-\frac{\nu_2}{\sigma^2}-\frac{1}{2\sigma^2}\sum_{t=1}^{N}(y_t-f(\ _t))^2\\&-\frac{1}{2}\sum_{i=0}^{p}I_{\alpha i}(\log{\sigma_\alpha}^2+\frac{{\alpha_i}^2}{{\sigma_\alpha}^2})-\frac{1}{2}\sum_{j=1}^{M}I_{\beta_j}\Phi(\sum_{i=0}^{p}I_{\gamma_{ji}})(\log{\sigma_\beta}^2+\frac{{\beta_j}^2}{{\sigma_\beta}^2})\\&-\frac{1}{2}\sum_{j=1}^{M}\sum_{i=0}^{p}I_{\beta_j}I_{\gamma_{ji}}\times(\log{\sigma_\gamma}^2+\frac{{\gamma_{ji}}^2}{{\sigma_\gamma}^2})-\frac{m}{2}\log(2\pi)+m\log\lambda-\log(m!)\end{aligned}$$

8- 6

In order to effectively implement the BNNs based on the posterior probability distributions defined above, the data preparation and hyperparameter settings suggested by Liang (2005b) were adopted in this study. Firstly, the input and output data were normalized by $q_t^{'}=(q_t-\overline{q})/S_q$, where $\overline{q}$ and S_q denote the mean and standard deviation of the input and output data. This type of data processing tends to avoid that the neural networks are trained to accommodate different scales of the observed data. For the settings of the hyperparameters, moderate values were adopted to penalize a

large weight variation and complex model structure: σ_α^2, σ_β^2, and σ_γ^2 are set to 5, and a vague prior on σ^2 was chosen through setting $\nu_1 = \nu_2 = 0.05$.

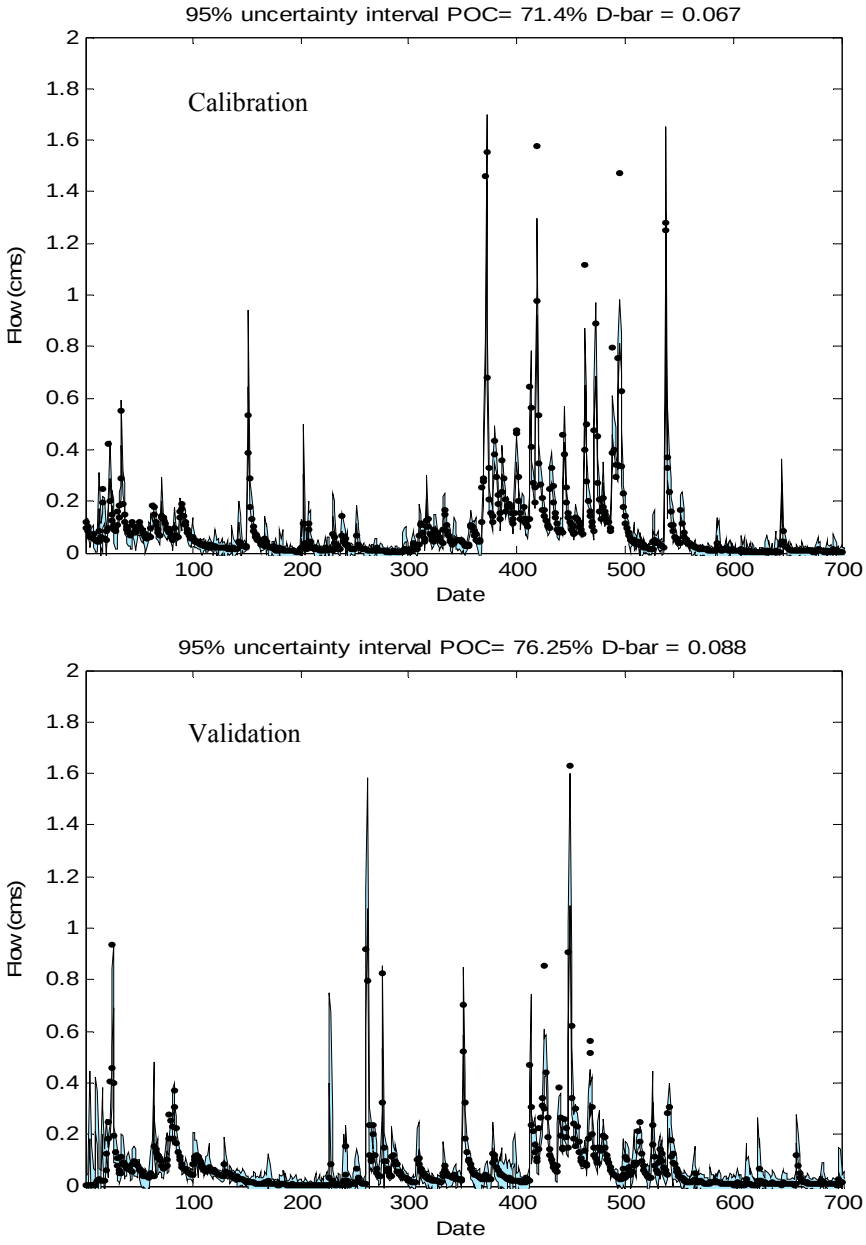

Figure 8-1. 95% uncertainty intervals estimated by BNN for the calibration and validation period in MCEW.

8.3 Streamflow uncertainty estimation using SWAT and BNN

The results obtained in Chapter VII show that the uncertainty interval obtained through applying SWAT and EMC with consideration of parameter uncertainty can only include about 20% observed data, which is far from the desired 95%. In this study, a BNN model was developed for the MCEW for uncertainty estimation of streamflow simulation. Streamflow data from water years (WY) 1997-1998 were used for neural network training. The second group included the streamflow data in WY 1999-2000, which were used to test the generalizing ability of trained networks. Model setup of the neural network involves the selection of input variables and hidden units. The input variables for the first layer of a three-layer perceptron network were selected based on the knowledge of the hydrologic characteristics of the study area and the correlation between the input variables and streamflow data. A total of 11 input variables were selected: 1) total daily precipitation of the last four days starting from $t-1$ to $t-3$ was taken into account as four separate inputs; 2) moving average of the last 30 days' precipitation as a single separate input; 3) mean daily temperature of the last four days starting from $t-1$ to $t-3$ were taken as four separate inputs; 4) moving average of last 30 days' temperature as one input; 5) daily streamflow values of the last three days from $t-1$ to $t-3$ were included according to the partial auto-correlation function (PACF). The selection of the number of hidden units was based on a trial and error procedure, and twenty hidden units were chosen for this case study. The EMC algorithm was implemented to train the BNNs using the same setting of the posterior probability defined in section 7.3.2.2. For training the Bayesian neural networks, the EMC was run for 200,000 iterations each time. The trace of the mean log posterior density was inspected, and it was found that convergence was reached after about 100,000 iterations. For each run of EMC-based BNNs, the first 100,000 iterations were taken as a burn-in stage and were discarded, and 10,000 sets of (θ,Λ) separated with equal interval were sampled from the remaining 100,000 iterations. A total of 50,000 samples was collected to derive the posterior distribution of (θ,Λ). The 50,000 sets of (θ,Λ) were used to run the neural network and calculate the network output $f(\mathbf{x},\theta,\Lambda)_k$ $(k=1,2,\ldots,50000)$. The

mean of the simulations of the 50,000 $f(\mathbf{x},\theta,\Lambda)$ was used as the estimate of the streamflow value. All the 50,000 $f(\mathbf{x},\theta,\Lambda)$ predictions were ranked in ascending order to determine the 95% prediction interval.

Figure 8-1 shows the 95% uncertainty intervals estimated by BNN for the calibration and validation period in MCEW. In Figure 8-1, Shaded are denote the 95% uncertainty interval, solid circle denotes observed data. The POC coefficients show that the BNN model can include more than 70% of the observed data, which is much higher than that obtained by the SWAT model. Neural network has powerful capacity of capturing non-linear relationships between inputs and outputs data without requiring an in-depth understanding of the underlying physical processes (Kingston et al., 2005), while SWAT try to simulate the hydrological processes using physically based equations. Theses two models are different in the underlying assumptions about the model structure. This, to some extent, indicates that the uncertainty associated with model structure can exert substantial impact on the uncertainty estimation of hydrologic modeling. But it is also worth noting that the uncertainties of SWAT modeling are also contributed by the errors from many input data (e.g. soil, land cover, elevation). It is difficult to identify the contributions from these input data and model parameters and structure without detailed information for the characteristics of theses uncertainty sources.

8.4 Streamflow uncertainty estimation using different types of BNNs

In this investigation, we are interested in the modeling and predictive uncertainty limits of streamflow simulation using BNNs with different treatment of uncertainties associated with parameters and structures of neural networks. Four types of BNNs were applied in this study. The first type of BNN, referred to as BNN-a, is with a fixed model structure and non-informative prior knowledge of parameters. The second one is BNN-b, which uses variable model structure and non-informative prior knowledge of parameters and model structures. The form of the posterior distribution used by BNN-a and BNN-b is defined by equation (7-11). The other two BNNs are referred to as BNN-c and BNN-d, respectively. BNN-c is with fixed model structure and informative prior knowledge of parameters. BNN-d uses variable model structure and informative prior knowledge of

parameters and model structures. The form of the posterior distribution used by BNN-c is a simplification of equation (8-4) through setting all the indicator functions equal to 1. The form of the posterior distribution used by BNN-d is defined by equation (8-6). Several comparison scenarios were designed to show the effect of taking variable model structure and informative prior knowledge into account on the uncertainty limits estimation using BNNs: 1) comparing the uncertainty limits obtained by BNN-a and BNN-b can provide insight into the effect of considering model structure uncertainty under the non-informative prior knowledge condition; 2) comparing BNN-c and BNN-d can show the effect of considering variable model structure under the informative prior knowledge condition; 3) comparing BNN-a and BNN-c can reveal the effect of considering prior knowledge under the fixed model structure condition; 4) comparing BNN-b and BNN-d can show the effect of considering prior knowledge under the variable model structure condition. These comparisons are expected to provide some insight into response of uncertainty limits due to different considerations of uncertainties related to parameters and model structure.

8.4.1 Application of BNNs in the RCEW

Streamflow during low temperature periods (late fall, winter, and early spring) in the RCEW is mainly driven by snowmelt. Because simulation of the snow-driven flow during these low temperature periods requires long term climate inputs, stremflow simulation during these periods was not included in the data sets for this study. Streamflow data from day 148 to 274 for water years (WY) 1968-1975 (a total of 1016 data values) were used in this study. These 1016 data values were further subdivided into two groups. The first group included the streamflow data in WY 1971-1972, which were used for neural networks training. The second group included the streamflow data in WY 1968-1970, which were used to test the generalizing ability of trained networks. A total of 13 input variables were selected: 1) total daily precipitation of the last four days starting from $t-1$ to $t-4$ was taken into account as four separate inputs; 2) moving average of the last 30 days' precipitation as a single separate input; 3) mean daily temperature of the last four days starting from $t-1$ to $t-4$ were taken as four separate

inputs; 4) moving average of last 30 days' temperature as one input; 5) daily streamflow values of the last three days from $t-1$ to $t-3$ were included according to the partial auto-correlation function (PACF). Thirty hidden units were used to determine the neural network structure of the ANNs and BNNs. The EMC algorithms were used to generate 50,000 sets of (θ, Λ), and estimate the 95% uncertainty intervals of the four types of BNNs.

The calibration and validation results of the BNNs are listed in Table 8-1. In terms of the MSE and R^2 coefficients for the validation period, the four BNNs show similar generalization performance with slight difference. For the four types of BNNs, we are particularly interested in the modeling and predictive uncertainty limits. For illustrative purposes, the modeling uncertainty intervals estimated by different BNNs for days from May 28, 1975 to July 12, 1975 during calibration period are shown in Figure 8-2. Similarly the predictive uncertainty intervals for days from May 28, 1972 to June 28, 1972 during validation period are shown in Figure 8-3. The two coefficients (POC and D-bar) used to evaluate the modeling and predictive uncertainty limits obtained by the four BNNs are shown in Table 8-1. For the validation period, BNN-a, BNN-b, BNN-c, and BNN-d produced POC coefficients of 65.83%, 80%, 93.23%, and 93.70%, respectively. And for validation period, BNN-a, BNN-b, BNN-c, and BNN-d produce POC coefficients of 73.75%, 83.46%, 93.70%, and 93.96%, respectively. We evaluated the 95% uncertainty intervals estimated by the four BNNs to show the effect of different considerations of uncertainty sources: 1) BNN-b includes about 10% more observations than BNN-a for both modeling and predictive uncertainty limits, which shows that considering variable model structures can improve the estimation of uncertainty limits under the non-informative prior knowledge condition; 2) under the informative prior knowledge condition, BNN-d yields 95% modeling and predictive uncertainty intervals with slightly better POC coefficients than BNN-c, while the D-bar coefficient of BNN-c are slightly smaller than BNN-d. In this case, variable model structures don not necessarily mean better uncertainty estimation of streamflow simulation 3) Under both fixed and variable model structure conditions, incorporating informative prior

knowledge of uncertainty sources can improve the uncertainty interval estimation to include more observations. The BNN-c includes about 28% and 20% more observations than BNN-a for modeling and predictive uncertainty intervals, respectively. BNN-d includes 10% more observations than BNN-b for both calibration and validation periods. Through comparing the uncertainty intervals of the four BNNs, it is evident that the choice of posterior model probability with different considerations of the uncertainties associated with parameters and structures can exert appreciable effects on the estimated uncertainty interval.

Table 8-1. Evaluation of the performance of the ANNs and BNNs for streamflow simulation in the RCEW.

Evaluation Coefficients / Period/Model		MSE	R^2	Percentage of coverage	Average interval width
Calibration	BNN-a	0.0055	0.98	65.83%	0.05
	BNN-b	0.0046	0.99	80.00%	0.08
	BNN-c	0.0058	0.98	93.23%	0.16
	BNN-d	0.0061	0.98	93.70%	0.17
Validation	BNN-a	0.0052	0.98	73.75%	0.06
	BNN-b	0.0050	0.98	83.46%	0.08
	BNN-c	0.0055	0.98	93.70%	0.14
	BNN-d	0.0056	0.98	93.96%	0.15

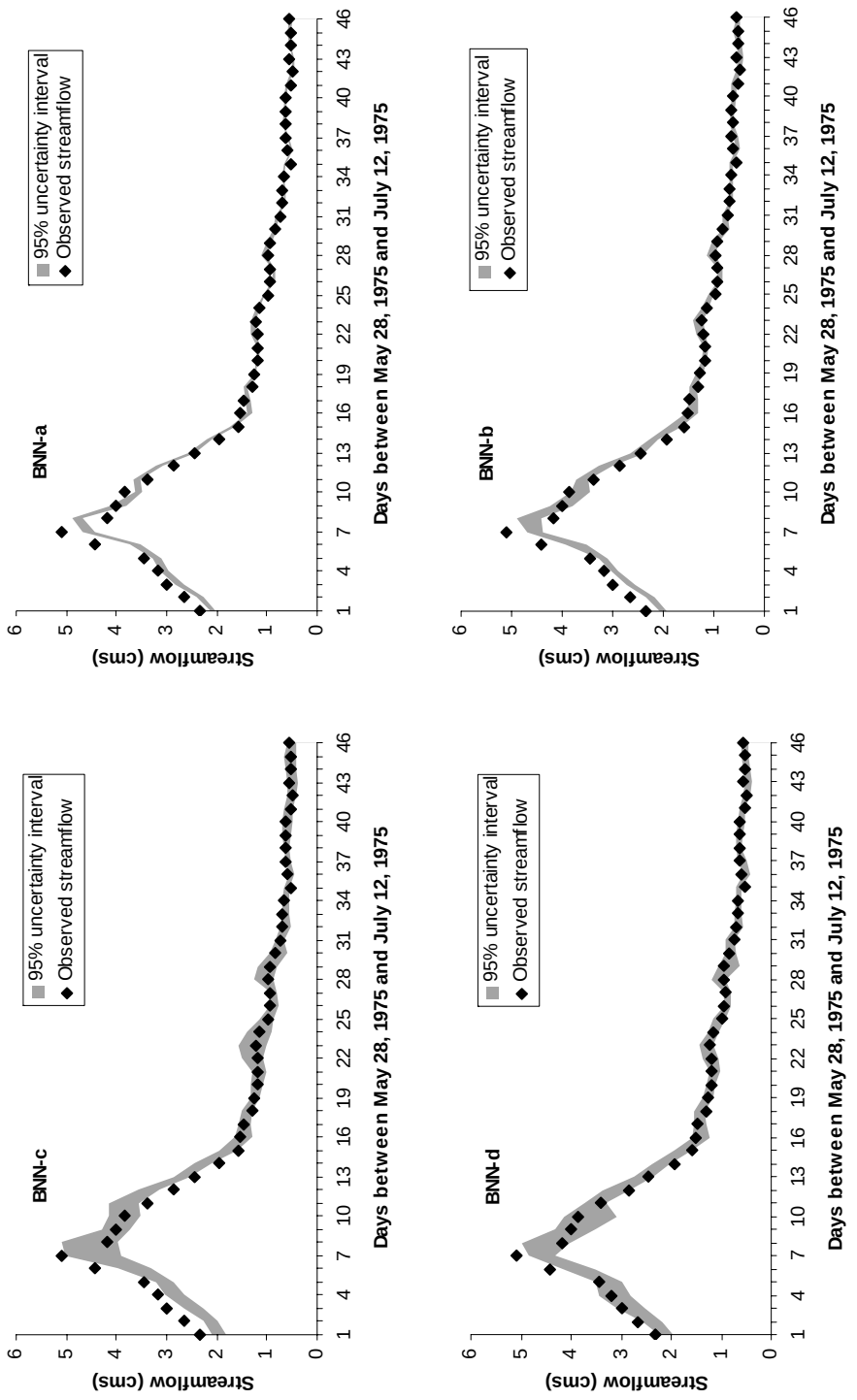

Figure 8-2. 95% modeling uncertainty intervals of streamflow simulation using different BNNs for days between May 28, 1975 and July 12, 1975 in RCEW

8.4.2 Application of BNNs in the LREW

Streamflow data of WY 1997-2002 in the LREW were used to develop and validate the BNNs and ANNs. These five years of daily data values were subdivided into two groups. The first group includes the streamflow values in WY 1997-2000, which were used for neural networks training. The second group includes the streamflow values in WY 2001-2002, which were used to test the generalizing ability of trained networks. The procedures for model setup of the neural network for the LREW are similar to those used in the RCEW. A total of 15 input variables were identified: 1) daily precipitation of the last five days starting from $t-1$ to $t-5$ was taken into account as five separate inputs; 2) moving average of the last 30 days' precipitation as a single separate input; 3) mean daily temperature of the last four days starting from $t-1$ to $t-5$ were taken as five separate inputs; 4) moving average of the last 30 days' temperature as one input; 5) daily streamflow values of the last three days from $t-1$ to $t-3$ were included as three separate according to the partial auto-correlation function (PACF). Thirty hidden units were used to determine the neural network structure of the ANNs and BNNs. The EMC algorithms were used to generate 50,000 sets of (θ, Λ), and estimate the 95% uncertainty intervals of the four types of BNNs.

Further analysis of the modeling and predictive uncertainty intervals estimated by the four BNNs in the LREW was also conducted. For illustrative purposes, the 95% modeling uncertainty intervals for days from January 4, 1997 to March 31, 1997 are shown in Figure 8-4 (calibration period), and the 95% predictive uncertainty intervals for days from January13, 2001 to April 24, 2001 are shown in Figure 8-5 (validation period). Visually, the difference between the uncertainty intervals estimated by the four BNNs in the LREW is not so appreciable compared with that obtained in the RCEW. The POC and D-bar values of the 95% uncertainty intervals estimated by different BNNs are listed in Table 8-2. For the calibration period, the BNN-a, BNN-b, BNN-c and BNN-d include about 79.71%, 82.29%, 85.14%, and 86% of the observed data into their 95% modeling uncertainty intervals respectively. For the validation period, the 95% predictive uncertainty intervals of the four BNNs tend to expand and include more observed data.

BNN-a, BNN-b, BNN-c, and BNN-d cover approximately 87.35%, 90.59%, 90.88%, and 92.06% of the observed streamflow data within their 95% predictive uncertainty intervals, respectively. The comparisons between the uncertainty intervals estimated by different BNNs in the LREW also show that incorporating variable model structures and informative prior knowledge can provide more reasonable estimation of the uncertainty of streamflow simulation: 1) under both variable and fixed model structure conditions, taking informative prior knowledge into account results in more robust modeling and predictive uncertainty intervals for BNN-d and BNN-c than BNN-b and BNN-a, respectively; 2) under both non-informative and informative prior knowledge conditions, BNN-b and BNN-d contain more observations in the modeling and predictive uncertainty limits than BNN-a and BNN-c respectively. Results obtained for the LREW are similar to those obtained for the RCEW, except that BNN-d produced not only larger POCs but also smaller D-bar values than BNN-c. This finding emphasizes the importance of considering multiple model structures.

Table 8-2. Evaluation of the performance of BNNs and ANNs for streamflow simulation in the LREW.

Evaluation Coefficients / Period/Model		MSE	R^2	Percentage of coverage	Average interval width
Calibration	BNN-a	11.25	0.94	79.71%	5.7
	BNN-b	11.17	0.94	82.29%	5.99
	BNN-c	12.16	0.93	85.14%	6.88
	BNN-d	13.14	0.93	86.00%	6.62
Validation	BNN-a	6.16	0.89	87.35%	5.32
	BNN-b	5.78	0.90	90.59%	5.59
	BNN-c	5.92	0.89	90.88%	6.58
	BNN-d	5.96	0.89	92.06%	5.8

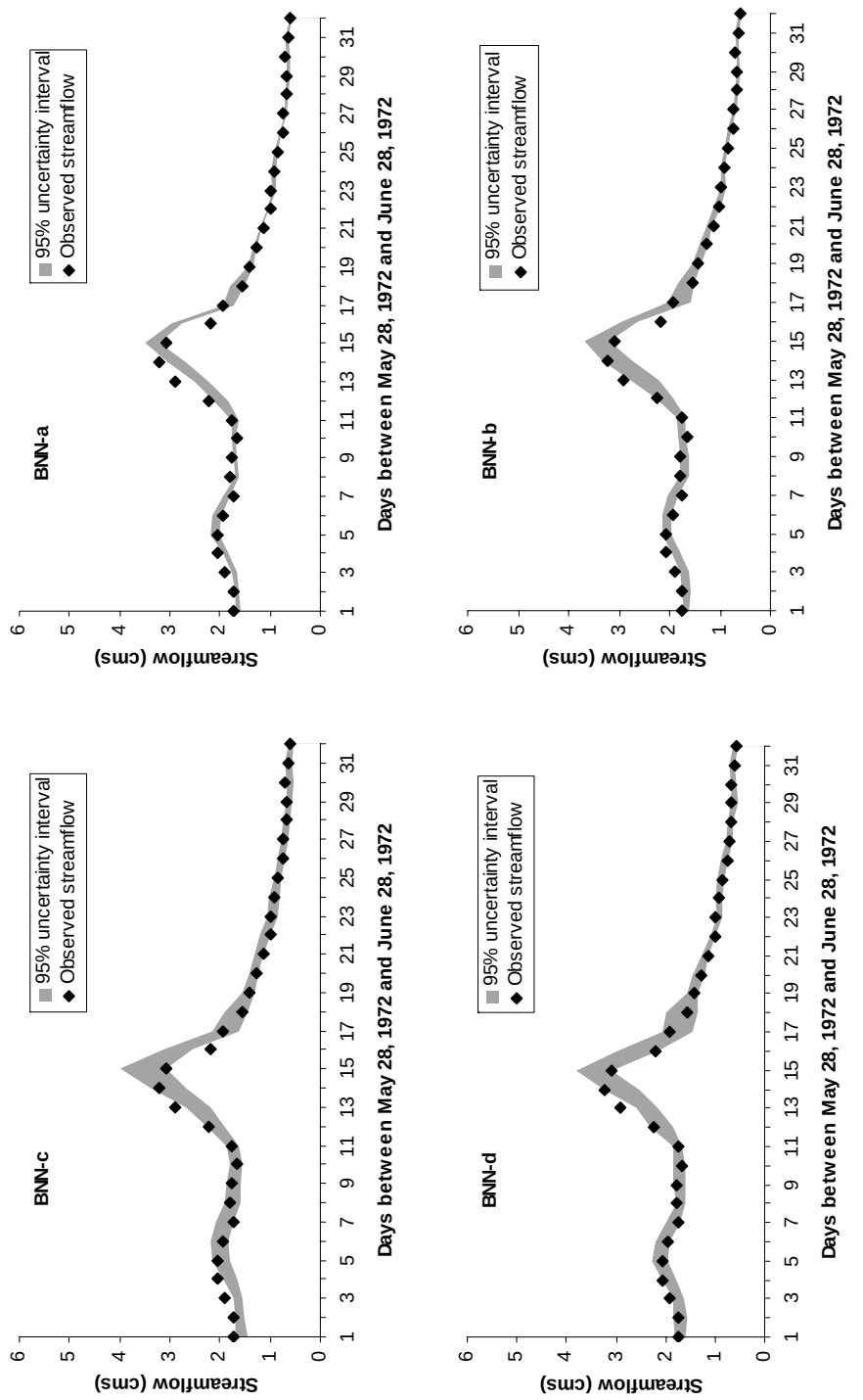

Figure 8-4. 95% predictive uncertainty intervals of streamflow simulation using different BNNs for days between May 28, 1972 and June 28, 1972 in RCEW.

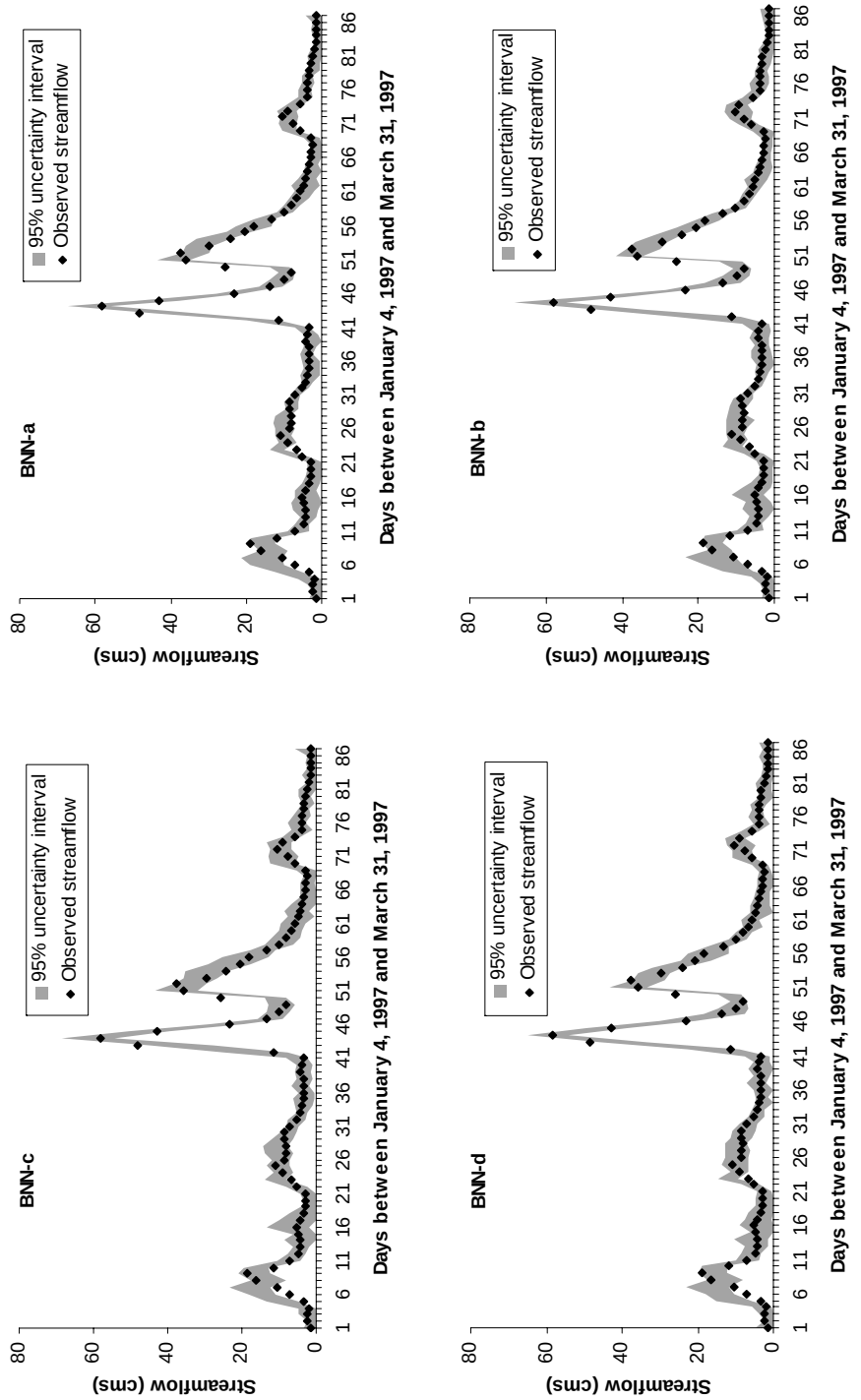

Figure 8-6. 95% modeling uncertainty intervals of streamflow simulation using different BNNs for days between January 4, 1997 and March 31, 1997 in LREW

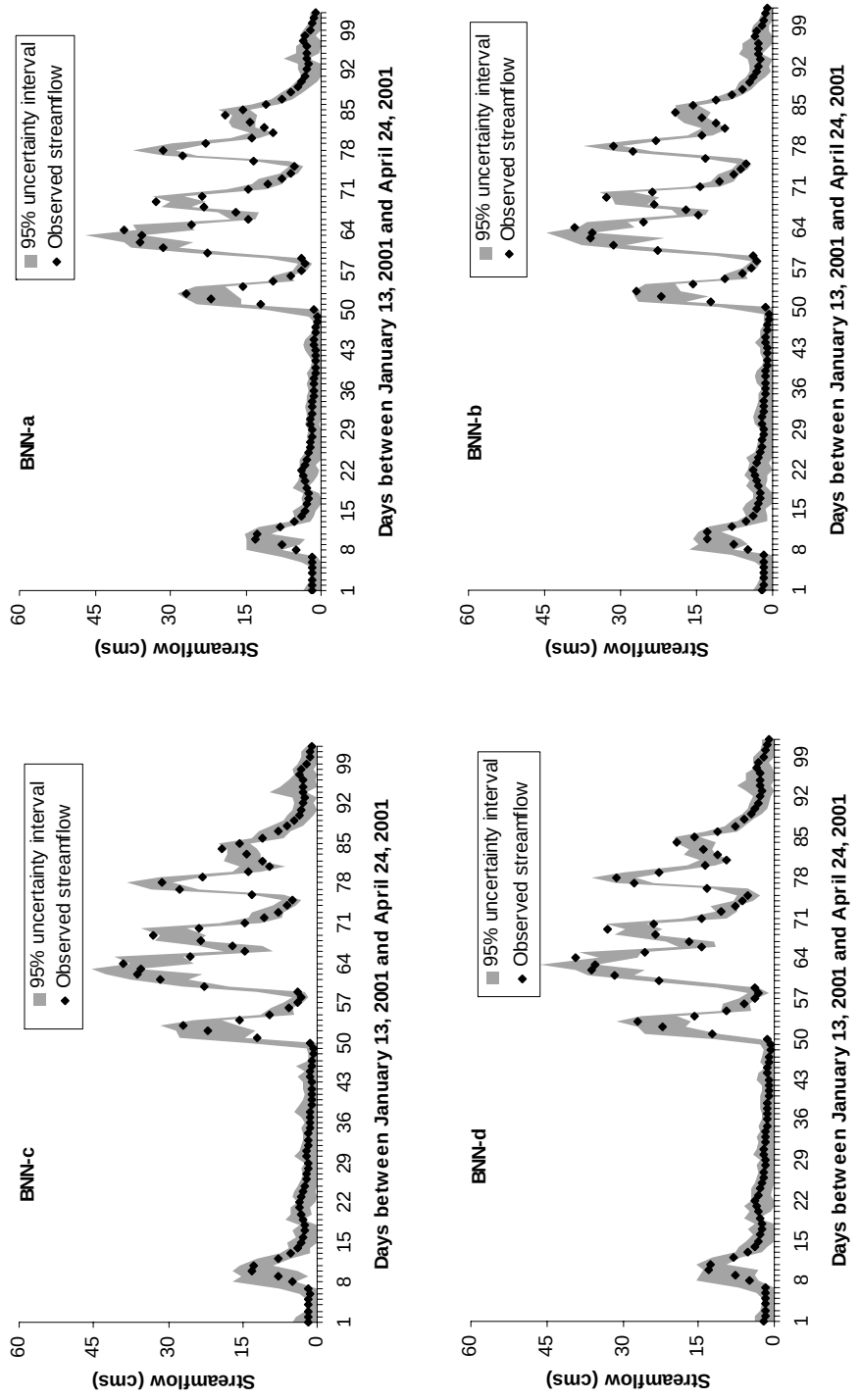

Figure 8-8. 95% predictive uncertainty intervals of streamflow simulation using different BNNs for days between January13, 2001 and April 24, 2001 in LREW.

8.4.3 Discussion

For the BNNs, the choice of posterior model probability with different considerations of uncertainties associated with model structures and parameters can exert substantial impacts on both modeling and predictive uncertainty limits estimated by the BNNs. Based on the test results in the LREW and the RCEW, the BNN-a, which only considers parameter uncertainty with non-informative prior knowledge, performs the least among all the four BNNs. On the other hand the BNN-d, which considers both parameter and model structure uncertainties with informative prior knowledge, produces equivalent or better estimation of the 95% modeling and predictive uncertainty intervals compared to the other BNNs. In general, incorporating variable model structure and informative prior knowledge produces more reasonable uncertainty interval estimation. It is important to recognize that the prior knowledge of neural networks' parameters and structures applied in this study is not selected arbitrarily, but based on expert knowledge and experimental testing (Wang, 1995, Muler and Insua, 1998, Liang, 2005b). An inappropriate setting of the prior knowledge may lead to worse estimation results.

From Figures 8-2 to 8-5 and Tables 8-1 to 8-2, we note that no uncertainty bounds estimated by the four BNNs can include 95% or more of the observed streamflow data, although BNN-c and BNN-d can produce POC coefficients approaching 95%. This is mainly because our understanding of hydrologic uncertainty is still far from complete. The inappropriate convergence to the true posterior (Kingston et al., 2005), the inadequate definition of the prior distribution of parameters and model structures, and the omission of uncertainties related to the observed input data and other forcing data can lead to inappropriate estimation of the uncertainties. Moreover, the complex and true joint distributions of the uncertainty sources (which result from high non-linearity of the hydrologic system and the complex interactions between different components of the system) make it very difficult to accurately represent the uncertainty of streamflow simulation (Liu and Gupta, 2007).

For water resources investigations essential for relevant decision making processes, the predictive uncertainty estimation of hydrologic prediction is valuable. The predictive

uncertainty limits are dependent on and different from modeling uncertainty. This is because when the trained BNNs are applied to another set of data independent of the training data, the hydrologic conditions may change and therefore impact the predictive interval estimation. From Figures 8-2 to 8-4 and Tables 8-1 to 8-2, it can be seen that the 95% modeling uncertainty limits are always narrower than the corresponding predictive uncertainty limits estimated by the same BNNs. The difference between modeling and predictive uncertainty limits can be impacted by the type of BNN and the characteristics of the hydrologic conditions. For example, in the RCEW, the BNN-a's POC of predictive uncertainty interval (73.75%) is about 8% higher than its POC of modeling uncertainty interval (65.83%), while the BNN-d's POC of predictive uncertainty interval (93.96%) is about the same its POC of modeling uncertainty interval (93.70%). Applying the BNN-d to the LREW, it is apparent that the BNN-d's POC of predictive uncertainty interval (92.06%) is 6% higher than its POC of modeling uncertainty interval (86.00%). Because of the future uncertainties due to natural and anthropogenic factors, the predictive uncertainty limits are also uncertain, which means that we are unable to estimate predictive uncertainty limits even if our estimation of modeling uncertainty limits are accurate. Hence in application of uncertainty analysis for hydrologic prediction, how to extend modeling uncertainty limits to predictive uncertainty limits remains a huge challenge for applying BNNs to water resources-related management and design problems.

Although uncertainty estimation of hydrologic prediction faces many challenges, it is still broadly recognized that proper consideration of uncertainty in hydrologic predictions is essential for purposes of both research and operational modeling (Wagener and Gupta, 2005; Pappenberger and Beven, 2006; Liu and Gupta, 2007). To improve the estimation of modeling uncertainty of hydrologic modeling, effective methods for considering the uncertainties associated with input hydrometeorolgic data (e.g. Kavetshi et al*,* 2006, Ajami et al., 2007, Srivastav et al., 2007) and observed outputs (e.g. Kuczera, 1983; Bates and Campbell, 2001; Yang et al., 2007) must also be considered in the definition of posterior model probability

8.5 Summary

In this study, the effect of model structure on the uncertainty estimation of streamflow simulation was examined. The SWAT model and Neural Network model were applied in MCEW for streamflow simulation of the same calibration and validation periods. The EMC method was used to derive the 95% uncertainty interval of streamflow simulation. The results show that the uncertainty intervals obtained by SWAT and Neural Network model are substantially different from each other. This, to some extent, indicates the model structure is an important source of uncertainty of hydrologic modeling. Future research on improving SWAT model structure and understanding the contributions of different forcing data uncertainties need to be evaluated.

Four types of BNNs with different treatments of variable structures and prior knowledge have been applied in this study. Findings from this study show that the 95% uncertainty limits of neural network outputs estimated by different BNNs were evidently different from each other. In general, BNNs incorporating multiple model structures can provide equal or better estimation of the uncertainty limits than those with fixed network structures. Findings also show that taking informative prior knowledge of network parameters and structures can lead to more robust estimation of the uncertainty limits. For all the test cases, the 95% uncertainty intervals (including modeling and predictive uncertainty intervals) estimated by all four BNNs failed to include 95% or more of observed streamflow data. This, to some extent, indicates the incomplete consideration of all uncertainty sources and inappropriate definition of error characteristics associated with different uncertainty sources. In the future, improving understanding and quantifying methods of different uncertainty sources need to be exploited for effective estimation of the uncertainty of hydrologic prediction using BNNs. It should also be noted that the difference between predictive uncertainty and modeling uncertainty, which is raised by unknown future conditions, complicates the process to develop practical guides on how to extend modeling uncertainty estimation to reliable predictive uncertainty estimation.

CHAPTER IX

CONCLUSIONS

In recent years, physically based distributed hydrologic models have been widely used by hydrologists and resources managers as tools to understand and manage natural and human activities that affect watershed systems. Hydrologic models, even those physically-based models, often contain parameters that cannot be measured directly due to measurement limits and scale issues (Beven, 2000; Madsen, 2003). In practical application of these models, parameters need to be estimated through an inverse method to reach agreement between observed and predicted output values. With the popularity of complex, distributed models, the time consumed for running these models is increasing substantially. Selecting efficient and effective parameter optimization algorithms for computationally expensive hydrologic models is becoming a nontrivial issue. Therefore, this study focuses on the developing and evaluating the efficiency and effectiveness of parameter calibration and uncertainty methods for a computationally intensive distributed hydrologic model. In this study we selected one complex distributed hydrologic model - SWAT as an example, which has been applied worldwide for hydrologic assessment.

To evaluate the efficacy of different optimization algorithms, five optimization algorithms (GA, SCE, PSO, DE and AIS), which have been successfully applied in optimization problems in different research fields, were tested for automatic parameter calibration the SWAT model in four watersheds with different terrain and climate conditions. The results show that no one optimization algorithm can consistently perform better than the other algorithms for the four watersheds. Based on the overall performance of the five optimization algorithms within limited model runs in the four watersheds, GA tends to find better objective function values given 10000 model

evaluations, while PSO can find acceptable good objective function values with less number of model evaluations (e.g. 1000 model evaluations). In order to find global optimum, multiple algorithms should be run to calibrate the SWAT model if the time and computational resources allow. Incorporating the strength of different algorithms into one powerful algorithm seems to hold promise for future investigations.

Although SWAT can simulate spatially distributed hydrologic variable, many applications of SWAT used single objective parameter optimization algorithms. In this study, the single objective optimization method and multi-objective optimization algorithm were applied to optimize the parameters of SWAT using observed streamflow data at various monitoring sites within the RCEW. The results obtained in this study show that the parameter solutions optimized using the objective function at one monitoring site performed worse than those obtained through simultaneously considering objective functions at multiple monitoring sites, which stresses the importance of collecting detailed spatially distributed data to calibrate the SWAT model. When using multi-site observed data to calibrate SWAT, the multi-objective optimization method can identify multiple Pareto optimal parameter solutions, which performed about the same as the parameter solutions obtained by the single objective optimization method for a calibration period and actually performed better for a validation period. The Pareto optimal parameter solutions can also be used to assess the uncertainty of simulated hydrographs. Overall, the multi-objective optimization method provides promising results that can be used for multi-site calibration of SWAT model.

Different multi-objective optimization algorithms were evaluated for simultaneously optimizing several objectives (e.g. multiple criteria, multiple flow components, and hydrologic variables at multiple sites) of SWAT. The tested multi-objective optimization algorithms include (SPEA2, NSGAII, ε-NSGAII, and three variants of MOPSO). It was found that the PSO based method converge quickly at the initial state, while GA based methods performed better in terms of finding good parameter sets with larger number of model runs. A new multi-objective optimization method (MO-PSOGA) that combines the advantages of the PSO and GA algorithms was proposed. Based on the evaluation of

the performances of different algorithms on three test cases, the MO-PSOGA method consistently perform better or close to the other algorithms.. The MO-PSOGA method can serve as a promising alternative method for multi-objective optimization of SWAT model.

In the application of the computationally intensive SWAT model, the time consumed by the parameter calibration is enormous. Using surrogate models to approximate the computationally intensive models is a promising method to save huge amounts of time for parameter calibration and uncertainty analysis. In this study, two learning machines (ANN and SVM) were evaluated and compared for approximating the SWAT model. It was found that both SVM and ANN can obtain high evaluation coefficients for approximating SWAT, however, SVM in general exhibited better generalization ability than ANN. In order to effectively and efficiently apply SVM to approximate SWAT, the effect of cross-validation schemes, parameter dimensions, and training sample sizes on the performance of SVM was evaluated and discussed. It is suggested that 3-fold cross-validation is adequate for training the SVM model, and reducing the parameter dimension through determining the parameter values from field data and the sensitivity analysis is an effective means of improving the performance of SVM. Simple examples were used to illustrate the potential applicability of combining the SVM model with uncertainty analysis and evolutionary optimization algorithm to save efforts for parameter calibration and uncertainty of SWAT.

Two types of parameter uncertainty analysis methods (GLUE and EMC) were applied to estimate simulation uncertainty of SWAT. Usually, the parameter uncertainty analysis needs more than 10000 model evaluations of SWAT. Considering the huge time consumed by applying parameter uncertainty analysis of SWAT, the Support Vector Machine (SVM) was used as a surrogate of SWAT to implement GLUE and EMC. The results show that the combination of GLUE with SVM (GLUE-SVM) can save more than 50% runs of the computationally intensive SWAT model compared with the implementation of the original GLUE method. At the same time, GLUE-SVM can provide uncertainty interval estimation close to that obtained by the original GLUE

method. As SVM is not a error free surrogate of SWAT, the combination of EMC and SVM can not obtain uncertainty interval estimation results similar to that obtained by the original EMC. It is worth noting that both GLUE and EMC can not accurately quantify the prediction uncertainty of SWAT model if parameter is taken as the only source of uncertainty.

The reasonable estimation of the prediction uncertainty, a valuable for decision making to address water resources management and design problems, is influenced by the techniques used to deal with different uncertainty sources. In this study, the effect of model structure on the uncertainty estimation of streamflow simulation was examined through applying SWAT and Neural Network models in MCEW. The 95% uncertainty intervals estimated by SWAT can only include 20% observed data, which Neural Network can include more than 70%. This indicates the model structure is an important source of uncertainty of hydrologic modeling and need to be evaluated carefully. Further exploitation of the effect of different treatments of the uncertainties of model structures on hydrologic modeling was conducted through applying four types of BNNs. The BNNs that only consider the parameter uncertainty with non-informative prior knowledge contain the least number of observed streamflow data in their 95% uncertainty bound. By considering variable model structure and informative prior knowledge, the BNNs can provide more reasonable quantification of the uncertainty of streamflow simulation. This study stresses the need for improving understanding and quantifying methods of different uncertainty sources for effective estimation of uncertainty of hydrologic simulation.

In this dissertation, different parameter calibration and uncertainty methods were evaluated or developed in order to efficiently and effectively to estimate parameters for computationally distributed hydrologic model. These methods were programmed with user friendly interface, which facilitate their easy use for SWAT and other models. As the results obtained show that that only consider the parameter uncertainty is not adequate to estimate the uncertainty of hydrologic simulation, future research is stressed

to be conducted on taking different uncertainty sources into account for effective estimation of uncertainty of hydrologic simulation.

REFERENCES

Abbaspour, K. C., C. A. Johnson, and M. T. van Genuchten. 2004. Estimating uncertain flow and transport parameters using a sequential uncertainty fitting procedure. *Vadose Zone J.* 3(4):1340-1352.

Ajami, N. K., Q. Duan, and S. Sorooshian. 2007. An integrated hydrologic Bayesian multimodel combination framework: Confronting input, parameter, and model structural uncertainty in hydrologic prediction. *Water Resour. Res.* 43, W01403, doi:10.1029/2005WR004745.

Ajami, N. K., H. V. Gupta, T. Wagener, and S. Sorooshian. 2004. Calibration of a semi-distributed hydrologic model for streamflow estimation along a river system. *J. Hydrol.* 298:112-135.

Ali, M. M., C. Khompatraporn, and Z. B. Zabinsky. 2005. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *J. Global Optim.* 31:635-672.

Almasri, M. N., and J. J. Kaluarachchi. 2005. Multi-criteria decision analysis for the optimal management of nitrate contamination of aquifers. *J. Envir. Manage.* 74(4):365-381.

Aly, A. H., and R. C. Peralta. 1999. Optimal design of aquifer cleanup systems under uncertainty using a neural network and a genetic algorithm. *Water Resour. Res.* 35(8): 2523-2532.

Andrieu, C., N. D. Freitas, and A. Doucet. 2001. Robust full Bayesian learning for radial basis networks. *Neural Comp.* 13, 2359-2407.

Arabi, M, R. S. Govindaraju, and M. M. Hantush. 2006. Cost-effective allocation of watershed management practices using a genetic algorithm. *Water Resour. Res.* 42:W10429, doi:10.1029/2006WR004931.

Arnold, J. G., P. M. Allen, and G. Bernhardt. 1993. A comprehensive surface groundwater flow model. *J. Hydrol.* 142:47-69.

Arnold, J. G., R. Srinivasan, R. S. Muttiah, and P. M. Allen, 1999. Continental scale simulation of the hydrologic balance. *J. Amer. Water Resour. Assoc.* 35(5):1037-1052.

Arnold, J. G., R. Srinivasan, R. S. Muttiah, and J. R. Williams, 1998. Large area hydrologic modelling and assessment part I: model development. *J. Amer. Water Resour. Assoc.* 34(1):73-89.

Arnold, J. G., J. R. Williams, and D. R. Maidment. 1995. Continuous-time water and sediment-routing model for large basins. *J. Hydraulic Eng.* 121(2):171-183.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. 2000a. Artificial neural networks in hydrology. I: preliminary concepts. *J. Hydrol. Eng.* 5(2): 115-123.

ASCE Task Committee on Application of Artificial Neural Networks in Hydrology. 2000b. Artificial neural networks in hydrology. II: hydrologic applications. *J. Hydrol. Eng.* 5(2): 124-137.

Babu, B. V., and K. K. N. Sastry. 1999. Estimation of heat transfer parameters in a trickle-bed reactor using differential evolution and orthogonal collocation. *Comp. Chem. Eng.* 23:327-339.

Bates, B. C., and E. P. Campbell. 2001. A Markov chain Monte Carlo scheme for parameter estimation and inference in conceptual rainfall-runoff modeling. *Water Resour. Res.* 37(4):937-947.

Bazi, Y., and F. Melgani. 2007. Semisupervised PSO-SVM regression for biophysical parameter estimation. *IEEE Geo. Remote Sens.* 45(6):1887-1895.

Bedient, P. B., W. C. Huber, and B. E. Vieux. 2001. *Hydrology and Floodplain Analysis (4th edition)*. Upper Saddle, NJ: Prentice Hall.

Bekele, G. E., and W. J. Nicklow. 2007. Multi-objective automatic calibration of SWAT using NSGA-II. *J. Hydrol.* 341:165-176.

Benaman, J. and C. A. Shoemaker. 2004. Methodology for analyzing ranges of uncertain model parameters and their impact on total maximum daily load processes. *J. Environ. Eng.* 130(6): 648-656.

Beven, K. J. 2000. *Rainfall-runoff Modelling:The Primer.* New York: John Wiley & Sons Press.

Beven, K. J. 2006. A manifesto for the enquiringly thesis. *J. Hydrol.* 320, 18-36.

Beven, K., and A. Binley 1992. The future of distributed models: model calibration and uncertainty prediction. *Hydrol. Process.* 6:279-298.

Beven, K. J., and J. Freer. 2001. Equifinality, data assimilation, and uncertainty estimation in mechanistic modeling of complex environmental systems. *J. Hydrol.* 249:11-29.

Bishop, C. M. 1995. *Neural Networks for Pattern Recognition.* New York: Oxford Univ. Press.

Brest, J., S. Greiner, B. Bošković, M. Mernik, and V. Žumer. 2006. Self-adapting control parameters in differential evolution: a comparative study on numerical Benchmark Problems. *IEEE Trans. Evol. Comput.* 10(6):646-656.

Byrski, A., and M. Kisiel-Dorohinicki. 2005. Immune-based optimization of predicting neural networks. *Computational Science - ICCS* 3516: 703-710.

Cao, W., B. W. Bowden, T. Davie, and A. Fenemor. 2006. Multi-variable and multi-site calibration and validation of SWAT in a large mountainous catchment with high spatial variability. *Hydrol. Process.* 20:1057-1073.

CEAP. 2008. *Conservation Effects Assessment Project.* Washington, D.C.: USDA Natural Resources Conservation Service. Available at: www.nrcs.usda.gov/technical/NRI/ceap/. Accessed on14 March 2008.

Chang. F. J., L. Chen, and L. C. Chang. 2005. Optimizing the reservoir operating rule curves by genetic algorithms. *Hydrol. Process.* 19, 2277-2289.

Chau, K. W. 2006. Particle swarm optimization training algorithm for ANNs in stage prediction of Shing Mun River. *J. Hydrol.* 329: 363-367

Chen, R., L. Pi, and C. Hsieh. 2005. Application of parameter optimization method for calibrating tank model. *J. Amer. Water Resour. Asso.* 41(2):389-402.

Chib, S. 1995. Marginal likelihood from the Gibbs output. *J. Amer. Stat. Assoc.* 90:1313-1321.

Cooper, V. A. , V. T. V. Nguyen, and J. A. Nicell. 1997. Evaluation of global optimization methods for conceptual rainfall-runoff model calibration. *Wat. Sci. Tech.* 36(5):53-60.

Cruz-Cortés, N, D. Trejo-Pérez, and C. A. Coello Coello. 2005. Handling constraints in global optimization using an artificial immune system. *LNCS* 3627: 234-247.

de Castro, L. N., and F. J. Von Zuben. 2002a. Learning and optimization using the clonal selection principle. *IEEE Trans. Evol. Comput.* 6(3): 239-251.

de Castro, L. N., and F. J. Von Zuben. 2002b. Automatic determination of radial basis functions: and immunity-based approach. *Inter. J. of Neur. Sys.* 11(6): 523-535.

de Freitas, J. F. G., M. Niranjan, A. H. Gee, and A. Doucet. 2000. Sequential Monte Carlo methods to train neural network models. *Neural Comput.* 12:955-993.

Deb, K. 2001. *Multi-objective Optimization Using Evolutionary Algorithms*. Chichester: John Wiley and Sons.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan. 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* 6(2): 182-197.

Demarty, J., C. Ottlé, I. Braud, A. Olioso, J. P. Frangi, H. V. Gupta, and L. A. Bastidas. 2005. Constraining a physically based soil-vegetation-atmosphere transfer model with surface water content and thermal infrared brightness temperature measurements using a multiobjective approach. *Water Resour. Res.* 41:W01011, doi:10.1029/2004WR003695.

Di Luzio M., R. Srinivasan, and J. G. Arnold. 2004. A GIS-coupled hydrological model system for the watershed assessment of agricultural nonpoint and point sources of pollution. *Trans. GIS* 8(1): 113-136.

Duan, Q., N. K. Ajami, X. Gao, and S. Sorooshian. 2007. Multi-model ensemble hydrologic prediction using Bayesian model averaging. *Adv. Water Resour.* 30(5):1371-1386.

Duan, Q., S. Sorooshian, and V. K. Gupta. 1992. Effective and efficient global optimization for conceptual rainfall-runoff models. *Water Resour. Res.* 28(4): 1015-1031.

Duan, Q. A., S. Sorooshian, and V. K. Gupta. 1994. Optimal use of the SCE-UA global optimization method for calibrating watershed models. *J. Hydrol.* 158: 265-284.

Eberhart, R. C., and Y. Shi. 1998. Comparison between genetic algorithms and particle swarm optimization. *LNCS* 1447: 611-616.

Eckhardt, K. and J.G. Arnold. 2001. Automatic calibration of a distributed catchment model. *J. Hydrol.* 251(1-2): 103-109.

Eckhardt, K., N. Fohrer, and H. G. Frede. 2005. Automatic model calibration. *Hydrol. Process.* 19:651-658.

Engeland, K., I. Braud, L. Gottschalk, and E. Leblois. 2006. Multi-objective regional modelling. *J. Hydrol.* 327:339-351

Ewen, J., G. O'Donnell, A. Burton, and E. O'Connell. 2006. Errors and uncertainty in physically-based rainfall-runoff modelling of catchment change effects. *J. Hydrol.* 330(3-4): 641-650.

Francos, A., F. J. Elorza, F. Bouraoui, G. Bidoglio, and L. Galbiati. 2003. Sensitivity analysis of distributed environmental simulation models: understanding the model behaviour in hydrological studies at the catchment scale. *Reliab. Eng. Syst. Safe.* 79(2): 205-218.

Freer, J., K. J. Beven and B. Ambroise. 1996. Bayesian estimation of uncertainty in runoff prediction and the value of data: an application of the GLUE approach. *Water Resour. Res.* 32(7): 2161-2173.

Gassman, P. W., M. Reyes, C. H. Green, and J. G. Arnold. 2007. The soil and water assessment tool: Historical development, applications, and future directions. *Trans. ASABE* 50(4):1212-1250.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin. 2004. *Bayesian Data Analysis(2nd Edition)*. Boca Raton, FL: Chapman & Hall/CRC.

Geyer, C. J. 1991. Markov chain Monte Carlo maximum likelihood, in *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, 153-163. Keramigas E. M., Fairfax, VA, eds. Seattle, WA.

Gill, M. K., T. Asefa, M. W. Kemblowski, and M. McKee, 2006a. Soil moisture prediction using support vector machines. *J. Amer. Water Resour. Assoc.* 42(4):1033-1046.

Gill, M. K., Y. H. Kaheil, A. Khalil, M. McKee, and L. Bastidas. 2006b. Multiobjective particle swarm optimization for parameter estimation in hydrology. *Water Resour. Res.* 42: doi:10.1029/2005WR004528.

Goldberg, D. 1989. *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading, MA: Addison-Wesley.

Govender, M. and C.S. Everson. 2005. Modelling streamflow from two small South African experimental catchments using the SWAT model. *Hydrol. Process.* 19(3): 683-692.

Green, P. J. 1995. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82, 711-732.

Gupta, H. V., S. Sorooshian, and P. O. Yapo. 1998. Toward improved calibration of hydrologic models: multiple and noncommensurate measures of information. *Water Resour. Res.* 34(4): 751-763.

Gupta. H. V., S. Sorooshian, and P. O. Yapo. 1999. Status of automatic calibration for hydrologic models: comparison with multilevel expert calibration. *J. Hydrol. Eng.* 4(2), 135-143.

Gutmann, H. M. 2001. A radial basis function method for global optimization. *J. Global Optim.* 19:201-227.

Haan, P.K. and R. W. Skaggs. 2003a. Effect of parameter uncertainty on DRAINMOD predictions I. Hydrology and yield. *Trans. of the ASAE* 46(4):1061-1067.

Haan, P.K. and R. W. Skaggs. 2003b. Effect of parameter uncertainty on DRAINMOD predictions II: Nitrogen Loss. *Trans. of the ASAE* 46(4): 1069-1075.

Haario, H., E. Saksman, and J. Tamminen. 2001. An adaptive metropolis algorithm. *Bernoulli* 7(2), 223-242.

Hastings, W. K. 1970. Monte-Carlo sampling methods using Markov Chains and their applications. *Biometrika* 57:97-109.

Higdon, D., H. Lee, and Z. Bi. 2002. A Bayesian approach to characterizing uncertainty in inverse problems using coarse and fine-scale information. *IEEE Transactions on Signal Processing* 50:389-399.

Holland, J. 1975. *Adaptation in natural and artificial systems.* Ann Arbor: University of Michigan Press.

Holmes, C. C. and B. K. Mallick. 1998. Bayesian radial basis functions of variable dimension. *Neural Comp.* 10:1217-1223.

Holvoet, K., A. van Griensven, P. Seuntjens, and P.A. Vanrolleghem. 2005. Sensitivity analysis for hydrology and pesticide supply towards the river in SWAT. *Phys. Chem.* Earth 30(8-10): 518-526.

Jha, M. K., A. Kumar, G. Nanda, and G. Bhatt. 2006. Evaluation of traditional and nontraditional optimization techniques for determining well parameters from step-drawdown test data. *J. Hydrol. Eng.* 11(6): 617-630.

Jia, Y., and T. B. Culver. 2006. Robust optimization for total maximum daily load allocations. *Water Resour. Res.* 42, W02412, doi:10.1029/2005WR004079.

Johnson, V. M., and L. L. Rogers. 2000. Accuracy of neural network approximators in simulation-optimization. *J. Water Resour. Plng. and Mgmt.* 126(2): 48-56.

Joshi, R, and A. C. Sanderson. 1999. Minimal representation multisensor fusion using differential evolution. *IEEE Trans. Syst. Man Cybernet. Part A* 29(1): 63-76.

Jung, B. S. and B. W. Karney. 2006. Hydraulic optimization of transient protection devices using GA and PSO approaches. *J. Water Resour. Plan. and Manage.* 132(1): 44-52

Kapelan, Z. S., and D. A. Savic. 2005. Multiobjective design of water distribution systems under uncertainty. *Water Resour. Res.* 41:W11407, doi:10.1029/2004WR003787.

Kavetski, D., G. Kuczera, and S. W. Franks. 2006. Bayesian analysis of input uncertainty in hydrological modeling: 1. Theory. *Water Resour. Res.* 42: W03407, doi:10.1029/2005WR004368.

Kennedy, J., and R. C. Eberhart. 1995. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks IV*, 1942-1948. Piscataway, NJ: IEEE Press.

Kennedy, J, and R. C. Eberhart. 2001. *Swarm Intelligence*. San Mateo, CA: Morgan Kaufmann.

Khalil, A., M. N. Almasri, M. McKee, and J. J. Kaluarachchi. 2005, Applicability of statistical learning algorithms in groundwater quality modeling. *Water Resour. Res.* 41:W05010, doi:10.1029/2004WR003608.

Khu, S. T., and H. Madsen. 2005. Multiobjective calibration with Pareto preference ordering: An application to rainfall-runoff model calibration. *Water Resour. Res.* 41:W03004, doi:10.1029/2004WR003041.

Kingston, G. B., M. F. Lambert, and H. R. Maier. 2005. Bayesian training of artificial neural networks used for water resources modeling. *Water Resour. Res.* 41:W12409, doi:10.1029/2005WR004152.

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Int. Joint Conf. Artif. Intell.* 14(2): 1137-1145.

Kollat, J. B. and P. Reed. 2005a. The value of online adaptive search: A comparison of NSGA-II, ε-NSGAII, and ε-MOEA. *LNCS* 3410: 386-398.

Kollat, J. B. and P. Reed.2005b. Comparing state-of-the-art evolutionary multi-objective algorithms for long-term groundwater monitoring design. *Adv. Water Resour.* 29:792-807.

Krishna, A. G. 2007. Optimization of surface grinding operations using differential evolution approach. *J. of Mater. Process. Tech.* 183(2-3): 202-029.

Kuczera, G. 1983. Improved parameter inference in catchment models: 1. evaluating parameter uncertainty. *Water Resour. Res.* 19(5):1151-1162.

Kuczera, G. 1997. Efficient subspace probabilistic parameter optimization for catchment models. *Water Resour. Res.* 33(1):177-185.

Kuo, S. F., and C. W. Liu. 2003. Simulation and optimization model for irrigation planning and management. *Hydro. Process.* 17:3141-3159.

Kuzera, G., and E. Parent. 1998. Monte Carlo assessment of parameter uncertainty in conceptual catchment models: The metropolis algorithm. *J. Hydrol.* 211:69-85.

Lampinen, J., and A. Vehtari. 2001. Bayesian approach for neural networks: Reviews and case studies. *Neural Networks* 14(3): 7-24.

Legates, D. R., and J. M. Gregory. 1999. Evaluating the use of 'goodness of fit' measures in hydrologic and hydroclimatic model validation. *Water Resour. Res.* 35(1): 233-241.

Lenhart, T., K. Eckhardt, N. Fohrer, and H. G. Frede. 2004. Comparison of two different approaches of sensitivity analysis. *Physics and Chemistry of the Earth* 27: 645-654.

Liang, F. 2003. An effective Bayesian neural network classifier with a comparison study to support vector machine. *Neural Comp.* 15:1959-1989.

Liang, F. 2005a. Evidence evaluation for Bayesian neural networks. *Neural Comp.* 17:1385-1410.

Liang, F. 2005b. Bayesian neural networks for non-linear time series forecasting. *Statis. Comp.* 15:13-29.

Liang, F. and Y. C. A. Kuk. 2004. A finite population estimation study with Bayesian neural networks. *Survey Methodology* 30:219-234.

Liang, F., and W.H. Wong. 2001a. Evolutionary Monte Carlo for protein folding simulations. *J. Chem Phys.* 115: 3374-3380.

Liang, F., and W.H. Wong. 2001b. Evolutionary Monte Carlo sampling: Applications to C_p model sampling and change-point problem. *Statistica Sinica* 10:317-342.

Liang, F., and W.H. Wong. 2001c. Real-parameter evolutionary sampling with applications in Bayesian mixture models. *J. Amer. Statist. Assoc.* 96:653-666.

Liong, S. Y., and C. Sivapragasam. 2002. Flood stage forecasting with support vector machines. *J. Amer. Water Resour. Assoc.* 38 (1): 173-196.

Liu, C. 2004. Study of some problems in water cycle changes of the Yellow River basin. *Advances in Water Sci. (in Chinese)* 15(5):608-614.

Liu, Y., and H. V. Gupta. 2007. Uncertainty in hydrologic modeling: Toward an integrated data assimilation framework. *Water Resour. Res.* 43:W07401, doi:10.1029/2006WR005756.

MacKay, D. J. C. 1992. A practical Bayesian framework for backprop networks. *Neural Comp.* 4:448-472.

Madsen, H. 2003. Parameter estimation in distributed hydrological catchment modelling using automatic calibration with multiple objectives. *Advances Water Res.* 26(2): 205-216.

Maier, H. R., and G. C. Dandy. 2000. Neural networks for the prediction and forecasting of water resources variables: A review of modeling issues and applications, Environ. *Model. Software* 15:101-123.

Marshall, L., D. Nott, and A. Sharma. 2004. A comparative study of Markov chain Monte Carlo methods for conceptual rainfall-runoff modeling. *Water Resour. Res.* 40:W02501, doi:10.1029/2003WR002378.

Massey, F. J. Jr. 1951. The Kolmogorov-smirnov test of goodness of fit. *J. Amer. Stat. Assoc.* 46(253): 68-78.

Metropolis, N., A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. 1953. equation of state calculations by fast computing machines. *Journal of Chemical Physics* 21:1087-1091.

Montanari, A., and A. Brath. 2004. A stochastic approach for assessing the uncertainty of rainfall-runoff simulations. *Water Resour. Res.* 40: doi:10.1029/2003WR002540.

Montanari, A., R. Rosso, and M. S. Taqqu. 1997. Fractionally differenced ARIMA models applied to hydrologic time series: Identification, estimation and simulation. *Water Resour. Res.* 33:1035-1044.

Moradkhani, H., K. L. Hsu, H. Gupta, and S. Sorooshian. 2005. Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter. *Water Resour. Res.* 41:W05012, doi:10.1029/2004WR003604.

Morshed, J., and J. J. Kaluarachchi. 1998. Parameter estimation using artificial neural network and genetic algorithm for free-product migration and recovery. *Water Resour. Res.* 34(5): 1101-1114.

Mugunthan, P., and C. A. Shoemaker. 2006. Assessing the impacts of parameter uncertainty for computationally expensive groundwater models. *Water Resour. Res.* 42, W10428, doi:10.1029/2005WR004640.

Muleta, M. K., and J. W. Nicklow. 2005. Sensitivity and uncertainty analysis coupled with automatic calibration for a distributed watershed model. *J. Hydrol.* 306:127-145.

Muller, P., and D.R. Insua. 1998. Issues in Bayesian analysis of neural network models. *Neural Comp.* 10:749-770.

Nash, J. E., and J. V. Sutcliffe. 1970. River flow forecasting through conceptual models, part I: a discussion of principles. *J. Hydrol.* 10(3): 282-290.

Neal, R. M. 1995. *Bayesian Learning for Neural Networks*. Ph.D. Dissertation. University of Toronto, Toronto, Ontario.

Neal, R. M. 1996. *Bayesian Learning for Neuraln Networks*. New York: Springer-Verlag.

Neitsch, S. L., A. G. Arnold, J. R. Kiniry, J. R. Srinivasan, and J. R. Williams. 2005a. *Soil and Water Assessment Tool User's Manual: Version 2005*. TR-192. College Station, Texas: Texas Water Resources Institute.

Neitsch, S. L., J. G. Arnold, J. R. Kiniry, K. W. King and J. R. Williams. 2005b. *Soil and Water Assessment Tool (SWAT) Theoretical Documentation*. Blackland Research Center, Texas Agricultural Experiment Station, Temple, TX.

Nelder, J.A., and R. Mead.1965. A simplex method for function minimization. *Comp. J.* 7: 308-313

Olivera, F., M. Valenzuela, R. Srinivasan, J. Choi, H. Cho, S. Koka, and A. Agrawal. 2006. ArcGIS-SWAT: A geodata model and GIS interface for SWAT. *J. Amer. Water Res. Assoc.* 42 (2): 295-309.

Pappenberger, F., and K. J. Beven. 2006. Ignorance is bliss: or seven reasons not to use uncertainty analysis. *Water Resour. Res.* 42:W05302, doi:10.1029/2005WR004820.

Parsopoulos, K. E., and M. N. Vrahatis. 2002. Recent approaches to global optimization problems through Particle Swarm Optimization. Natural Comp. 1:235-306.

Peschel, J. M., P. K. Haan, and R. E. Lacey. 2006. Influences of soil dataset resolution on hydrologic modeling. *J. Amer. Water Res. Assoc.* 42(5): 1371-1389.

Qi, C. and S. Grunwald. 2005. GIS-based hydrologic modeling in the sandusky watershed using SWAT. *Trans. ASAE* 48(1):169-180.

Rahman, T. K., A. Suliman, and S. I. Musirin. 2006. Artificial immune-based optimization technique for solving economic dispatch in power system. *Neural Network* 3931:338-345.

Reca, J., and J. Martínez. 2006. Genetic algorithms for the design of looped irrigation water distribution networks. *Water Resour. Res.* 42:W05416, doi:10.1029/2005WR004383.

Reddy, M. J. and D. N. Kumar. 2007. Multi-objective particle swarm optimization for generating optimal trade-offs in reservoir operation. *Hydrol. Process.* 21: 2897-2909.

Reed, P., B. S. Minsker, and D. E. Goldberg. 2003. Simplifying multiobjective optimization: an automated design methodology for the nondominated sorted genetic algorithm-II. *Water Resour. Res.* 39:1196, doi:1110.1029/2002WR001483.

Regis, R. G., and C. A. Shoemaker. 2005. Constrained global optimization using radial basis functions. *J. Global Optim.* 31(1):153-171.

Renard, B., V. Garreta, and M. Lang. 2006. An application of Bayesian analysis and Markov chain Monte Carlo methods to the estimation of a regional trend in annual maxima. *Water Resour. Res.* 42:W12422, doi:10.1029/2005WR004591.

Reyes Sierra, M., and C.A. Coello Coello. 2005. Improving PSO-based multi-objective optimization using crowding, mutation and ε-dominance. *LNCS* 3410: 505-519.

Reyes Sierra, M., and C.A. Coello Coello. 2006. Multi-objective particle swarm optimizers: a survey of the state-of-the-art. *Int. J. Comput. Intel. Res.* 2(3):287-308.

Santhi, C., J. G. Arnold, J. R. Williams, W. A. Dugas, and L. Hauck. 2001. Validation of the SWAT model on a large river basin with point and nonpoint sources. *J. Amer. Water Resour. Assoc.* 37(5):1169-1188.

Schaefli, B., D. B. Talamba, and A. Musy. 2007. Quantifying hydrological modeling errors through a mixture of normal distributions. *J. Hydrol.* 332: 303-315.

Schaffer, J. D., R. A. Caruana, L. J. Eshelman, and R. Das. 1989. A study of control parameters affecting online performance of genetic algorithms for function optimization. In *Proceedings of the Third International Conference on Genetic algorithms*, 51-60. Schaffer J. D., eds. Morgan Kaufmann: San Mateo, CA.

Scott, M., Uncertain models and modeling uncertainty. 2007. Available at http://www.nottingham.ac.uk/environmental-odelling/Marian%20Scott.ppt#261,8,Stages in modeling. Accesses on January 8, 2007.

Seyfried, M. S., R. C. Harris, D. Marks, and B. Jacob. 2000. *A geographic database for watershed research, Reynolds Creek Experimental Watershed*, Idaho, USA. USDA ARS Technical Bulletin NWRC-2000-3.

Sheridan, J. M. 1997. Rainfall-streamflow relations for coastal plain watersheds. *Trans. ASAE* 13(3): 333-344.

Shi, Y., and R. C. Eberhart. 1998. Parameter selection in particle swarm optimization. In *Evolutionary Programming VII*, *Porto VW*, 611-616. Saravanan N., Waagen D. and Eiben A. E., eds. Berlin: Springer.

Sloan, P. G., I. D. Morre, G. B. Coltharp, and J. D. Eigel. 1983. *Modeling surface and subsurface stormflow on steeply-sloping forested watersheds*, University of Kentucky, Lexington, Kentucky, Water Resources Institute Report 142.

Smola, A. J., and B. Scholköpf. 2004. A tutorial on support vector regression. *Stat. and Comput.* 14: 199-222.

Spruill, C. A., S. R. Workman, and J. L. Taraba. 2000. Simulation of daily and monthly stream discharge from small watersheds using the SWAT model. *Trans. ASAE* 43(6): 1431-1439.

Srivastav, R. K., K. P. Sudheer, and I. Chaubey. 2007. A simplified approach to quantifying predictive and parametric uncertainty in artificial neural network hydrologic models. *Water Resour. Res.* 43:W10407, doi:10.1029/2006WR005352.

Srivastava, P. 2002. Watershed optimization of best management practices using AnnAGNPS and a genetic algorithm. *Water Resour. Res.* 38(3): doi:10.1029/2001WR000365.

Storn R, and Price K. 1997. Differential evolution - A simple and efficient heuristic for global optimization over continuour spaces. *J. Global Optimiz.* 11:341-359.

Tang, Y., P. Reed, and T. Wagener. 2006. How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration? *Hydrol. Earth Syst. Sci.* 10:289-307.

Thiemann, M., M. Trosser, H. Gupta, and S. Srorooshian. 2001. Bayesian recursive parameter estimation for hydrologic models. *Water Resour. Res.* 37(10): 2521-2535

Tolson, B. A., and C. A. Shoemaker. 2007. Dynamically dimensioned search algorithm for computationally efficient watershed model calibration. *Water Resour. Res.* 43:W01413, doi:10.1029/2005WR004723.

Tolson, B. A., and C. A. Shoemaker. 2008. Efficient prediction uncertainty approximation in the calibration of environmental simulation models. *Water Resour. Res.* 44:W04411, doi:10.1029/2007WR005869.

van Griensven, A., and W. Bauwens. 2003. Multi-objective autocalibration for semi-distributed water quality models. *Water Resour. Res.* 39(12):1348, doi:10.1029/2003WR002284

van Griensven, A., and W. Bauwens. 2005. Application and evaluation of ESWAT on the Dender basin and Wister Lake basin. *Hydrol. Process.* 19(3): 827-838.

van Griensven, A., T. Meixner, S. Grunwald, T. Bishop, A. Diluzio, and R. Srinivasan. 2006. A global sensitivity analysis tool for the parameters of multi-variable catchment models. *J. Hydrol.* 324:10-23.

van Griensven, A., and T. Meixner. 2006. Methods to quantify and identify the sources of uncertainty for river basin water quality models. *Water Sci.Tech.* 53(1): 51-59.

Van Liew, M. W., and J. Garbrecht. 2003. Hydrologic simulation of the Little Washita River Experimental Watershed using SWAT. *J. Amer. Water Resour. Assoc.* 39(2):413-426.

Van Liew, M. W., J. G. Arnold, and D. D. Bosch. 2005. Problems and potential of autocalibrating a hydrologic model. *Trans. ASAE* 48(3): 1025-1040.

Van Liew, M. W., T. L. Veith, D. D. Bosch, and J. G. Arnold. 2007. Suitability of SWAT for the conservation effects assessment project: a comparison on USDA ARS watersheds. *J. Hydrol. Eng.* 12(2): 173-189.

Van Veldhuizen, D. A., and G. B. Lamont. 1998. *Multiobjective evolutionary algorithm research: A history and analysis*, Dept. Elec. Comput. Eng., Graduate School of Eng., Air Force Inst. Technol., Wright-Patterson AFB, OH, Tech. Rep. TR-98-03.

Vandenberghe, V., A. van Griensven, and W. Bauwens. 2001. Sensitivity analysis and calibration of the parameters of ESWAT: Application to the River Dender. *Water Sci. Technol.* 43(7): 295-301.

Vapnik, V. 1998. *Statistical Learning Theory*. New York: John Wiley and Sons.

Veith, T. L., A. N. Sharpley, J. L. Weld, and W. J. Grurek. 2005. Comparison of measured and simulated phosphorous losses with index site vulnerability. *Trans. ASAE.* 48(2):557-565.

Vrugt, J. A., and B. A. Robinson. 2007. Treatment of uncertainty using ensemble methods: Comparison of sequential data assimilation and Bayesian model averaging. *Water Resour. Res.* 43, W01411, doi:10.1029/2005WR004838.

Vrugt, J. A., C. G. H. Diks, H. V. Gupta, W. Bouten, and J. M. Verstraten. 2005. Improved treatment of uncertainty in hydrologic modeling: combining the strengths of global optimization and data assimilation. *Water Resour. Res.* 41:W01017, doi:10.1029/2004WR003059.

Vrugt, J. A., H. V. Gupta, L. A. Bastidas, W. Bouten, and S. Sorooshian. 2003a. Effective and efficient algorithm for multiobjective optimization of hydrologic models. *Water Resour. Res.* 39(8): doi:10.1029/2002WR001746.

Vrugt, J. A., H. V. Gupta, W. Bouten, and S. Sorooshian. 2003b. A shuffled complex evolution metropolis algorithm for optimization and uncertainty assessment of hydrologic model parameters. *Water Resour. Res.* 39(8):1201, doi:10.1029/2002WR001642.

Wagener, T., and H. V. Gupta. 2005. Model identification for hydrological forecasting under uncertainty. *Stoch. Environ. Res. Risk. Assess.* 19:doi:10.1007/s00477-005-0006-5.

Wagener, T., D. P. Boyle, M. J. Lees, H. S. Wheater, H. V. Gupta, and S. Sorooshian. 2001. A framework for development and application of hydrological models. *Hydrol. Earth Syst. Sci.* 5:13-26.

Wang, X., and A. M. Melesse. 2005. Evaluation of the SWAT model's snowmelt hydrology in a northwestern Minnesota watershed. *Trans. ASAE* 48(4): 1359-1376.

Wang, Y. 1995. Unpredictability of standard back propagation neural networks. *Manage. Sci.* 41:555-559.

White, K. L. and I. Chaubey. 2005. Sensitivity analysis, calibration, and validations for a multisite and multivariable SWAT model. *J. Amer. Water Resour. Assoc.* 41(5): 1077-1089.

Williams, J. R. and R.W. Hann. 1972. HYMO, a problem-oriented computer language for building hydrologic models. *Water Resour. Res.* 8(1):79-85

Wolpert, D. H., and W. G. Macready. 1997. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* 1(1): 67-82.

Yandamuri, S. R. M., and K. Srinivasan. 2006. Multiobjective optimal waste load allocation models for rivers using Nondominated Sorting Genetic Algorithm-II. *J. of Water Resour. Planning and Manage*. 132(3): 133-143.

Yang, J., P. Reichert, and K. C. Abbaspour. 2007. Bayesian uncertainty analysis in distributed hydrologic modeling: A case study in the Thur River basin (Switzerland). *Water Resour. Res.* 43:W10401, doi:10.1029/2006WR005497.

Yapo, P. O., H. V. Gupta, and S. Sorooshian. 1998. Multi-objective global optimization for hydrologic models. *J. Hydrol.* 204:83-97.

Ye, K. Q., W. Li, and A. Sudjianto. 2000. Algorithmic construction of symmetric Latin hypercube designs. *J. Stat. Plann. Inference* 90:145-159.

Yu, P., S. Chen, and I. Chang. 2006. Support vector regression for real-time flood stage forecasting. *J. Hydrol.* 328: 704-716.

Zak, S. K., and K. J. Beven. 1999. Equifinality, sensitivity and predictive uncertainty in the estimation of critical loads. *Sci. Total Environ.* 236(1-3): 191-214.

Zhang, X., R. Srinivasan, and F. Hao. 2007. Predicting hydrologic response to climate change in the Luohe River Basin using the SWAT model. *Trans. ASABE* 50(3): 901-910.

Zhang, X., R. Srinivasan, B. Debele, and F. Hao. 2008. Runoff simulation of the headwaters of the Yellow River using the SWAT model with three snowmelt algorithms. *J. Amer. Water Resour. Assoc.* 44(1):48-61.

Zitzler, E. and L. Thiele. 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. *IEEE Trans. Evol. Comp.* 3(4): 257-271.

Zitzler, E., L. Thiele, M. Laumanns, C. M. Fonseca, and V. Grunert da Fonseca. 2003. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans. Evol. Comput.* 7:117-132.

Zitzler, E., M. Laumannas and L. Thiele. 2002. SPEA2: Improving the strength pareto evolutionary algorithm for multi-objective optimization. Available at http://e-collection.ethbib.ethz.ch/ecol-pool/incoll/incoll_324.pdf. Accessed on August 8, 2007.

Zou, R., W. S. Lung, and J. Wu. 2007. An adaptive neural network embedded genetic algorithm approach for inverse water quality modeling. *Water Resour. Res.* 43:W08427, doi:10.1029/2006WR005158.

VITA

Name: Xuesong Zhang

Address: Spatial Sciences Laboratory
1500 Research Parkway, Suite B217
College Station, TX 77843-2120

Email Address: xuesongzhang2004@gmail.com

Education: B.S., Environmental Sciences, Qingdao University, China, 2001
M.S., Water Resources and Environment, Beijing Normal University, China, 2004
Ph.D, Watershed Management and Hydrologic Sciences, Texas A&M University, August, 2008

Major Awards & Honors

1. TAMU Academic Excellence Award (Richard E. Morris Endowed Memorial), Texas A&M University, 2007-2008.
2. Outstanding PhD Student, Department of Ecosystem Science and Management, Texas A&M University, 2006-2007.

Selected Journal Articles

1. Xuesong Zhang, Raghavan Srinivasan, and Fanghua Hao. Predicting Hydrologic Response to Climate Change in the Luohe River Basin Using the SWAT Model. Transactions of American Society of Biological and Agricultural Engineering, 2007, 50(3): 901-910.
2. Xuesong Zhang, Raghavan Srinivasan, Debele Bekele, and Fanghua Hao. Runoff Simulation of the Headwaters of the Yellow River Using the SWAT Model With Three Snowmelt Algorithms, Journal of American Water Resources Association, 2008, 44(1): 48-61.