ON THE PREDICTIVE UNCERTAINTY OF A DISTRIBUTED HYDROLOGIC MODEL

A Dissertation

by

HUIDAE CHO

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: Civil Engineering

ON THE PREDICTIVE UNCERTAINTY OF A DISTRIBUTED HYDROLOGIC MODEL

A Dissertation

by

HUIDAE CHO

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,	Francisco Olivera
Committee Members,	Anthony T. Cahill
	Seth D. Guikema
	Raghavan Srinivasan
Head of Department,	David V. Rosowsky

August 2008

Major Subject: Civil Engineering

ABSTRACT

On the Predictive Uncertainty of a Distributed Hydrologic Model. (August 2008)

Huidae Cho, B.S., Kyungpook National University;

M.S., Kyungpook National University

Chair of Advisory Committee: Dr. Francisco Olivera

We use models to simulate the real world mainly for prediction purposes. However, since any model is a simplification of reality, there remains a great deal of uncertainty even after the calibration of model parameters. The model's identifiability of realistic model parameters becomes questionable when the watershed of interest is small, and its time of concentration is shorter than the computational time step of the model. To improve the discovery of more reliable and more realistic sets of model parameters instead of mathematical solutions, a new algorithm is needed. This algorithm should be able to identify mathematically inferior but more robust solutions as well as to take samples uniformly from high-dimensional search spaces for the purpose of uncertainty analysis.

Various watershed configurations were considered to test the Soil and Water Assessment Tool (SWAT) model's identifiability of the realistic spatial distribution of land use, soil type, and precipitation data. The spatial variability in small watersheds did not significantly affect the hydrographs at the watershed outlet, and the SWAT model was not able to identify more realistic sets of spatial data. A new population-based heuristic called the Isolated Speciation-based Particle Swarm Optimization (ISPSO) was developed to enhance the explorability and the uniformity of samples in high-dimensional problems. The algorithm was tested on seven mathematical functions and outperformed other similar algorithms in terms of computational cost, consistency, and scalability. One of the test functions was the Griewank function, whose

number of minima is not well defined although the function serves as the basis for evaluating multi-modal optimization algorithms. Numerical and analytical methods were proposed to count the exact number of minima of the Griewank function within a hyperrectangle. The ISPSO algorithm was applied to the SWAT model to evaluate the performance consistency of optimal solutions and perform uncertainty analysis in the Generalized Likelihood Uncertainty Estimation (GLUE) framework without assuming a statistical structure of modeling errors. The algorithm successfully found hundreds of acceptable sets of model parameters, which were used to estimate their prediction limits. The uncertainty bounds of this approach were comparable to those of the typical GLUE approach.

To my family,

to my mother,

and in memory of my father

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to my adviser, Dr. Francisco Olivera, for his valuable advice, support, encouragement, and endless enthusiasm for the research. I gratefully acknowledge the committee members, Dr. Anthony T. Cahill, Dr. Seth D. Guikema, and Dr. Raghavan Srinivasan, for their support and taking their time to read this manuscript.

Many thanks also go to my colleagues who supported me in so many ways. Among them, I particularly wish to thank Dongkyun Kim for providing a great source of stimulation for the research. I am indebted to the Korea Ministry of Science & Technology and Texas A&M University for their financial support.

Finally, I am deeply grateful to my wife and children for their understanding, constant support, and encouragement, and to my mother and parents-in-law for their patience, love, and support. I dedicate this dissertation to the loving memory of my father.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

TABLE Page

LIST OF FIGURES

FIGURE Page

FIGURE Page

CHAPTER I

INTRODUCTION

The ever increasing capabilities of computers and software technologies have improved the collection of spatial data and enabled the development of distributed hydrologic models (Arnold et al., 1998; McIntyre et al., 2005). However, taking into account the distributed characteristics of the watershed does not necessarily improve model performance and lead to the better prediction of water quantity and quality variables at ungauged basins due to increasing model complexity. A large number of spatially distributed model parameters often cause the over-parameterization of the model structure (Koh et al., 2004; McIntyre et al., 2005), which results in the "equifinality" problem (Beven, 1993) where many different model parameter sets perform equally well. In addition, uncertainty associated with the measurement, the model parameters, and the model output is adding more difficulties to the application of distributed hydrologic models to ungauged basins.

We have to be very careful in choosing an appropriate model structure when accounting for spatial variability, such as the distribution of land use, soil type, and rainfall, in small watersheds where the time of concentration is shorter than the computational time step of the model. The effect of the spatial variability can be driven by the compatibility between the computational time step of the model and the time of concentration of the watershed. When everything happens in one time step of the model, the model may not be able to capture what is happening in the watershed in each time step. The Soil and Water Assessment Tool (SWAT) (Neitsch et al., 2002a) was used to investigate the effect of the spatial variability in small

---

This dissertation follows the style of Journal of Hydrology.

watersheds on the model output.

When the model is, at best, a simplification of reality, pursuing mathematically the best solution does not necessarily mean searching for a unique optimal solution Beven (2006a). There may exist even mathematically inferior solutions, often referred to as local optima, that provide more realistic predictions. However, it is not straightforward to find local optima using global optimization algorithms (Brits et al., 2007). This multi-modal nature of the model leads to the need for multi-modal optimization algorithms. A population-based heuristic algorithm called the Isolated Speciation-based Particle Swarm Optimization (ISPSO) was developed to find local optima in complicated and high-dimensional problem spaces.

The Griewank function (Griewank, 1981) is one of the mathematical functions used to test the new multi-modal optimization algorithm. The number of minima used by Brits et al. (2007) was found to be incompatible with the number of minima analytically and numerically found in this study. To the best of the author's knowledge, the number of minima of the Griewank function has not been analytically derived in the literature. To serve as the basis for evaluating algorithms designed to find local optima as well as global optima, the number of minima of the Griewank function was analytically derived by numerically restricting the domain space in hyperrectangles meeting certain requirements.

The Generalized Likelihood Uncertainty Estimation (GLUE) framework (Beven and Binley, 1992) has been widely used to assess the prediction uncertainty of the model (Schulz et al., 1999; Jacquin and Shamseldin, 2007). However, the lack of statistical assumptions for the error structure hinders the application of probabilistic sampling techniques, such as Markov chain Monte Carlo methods or importance samplings, to the GLUE framework. For this reason, random sampling is usually employed in the GLUE framework to find acceptable models, but is computationally

expensive (Beven, 2006a). The ISPSO algorithm was applied to the GLUE framework to reduce computational burden in uncertainty analysis, and the results for the proposed approach were compared with those for the typical GLUE approach.

The objectives of this study are four-fold: (1) to assess the effect of the spatial variability in small watersheds on the model output, (2) to develop an efficient multi-modal optimization algorithm for finding multiple solutions in multi-modal problems, (3) to analytically derive the number of minima of the Griewank function in numerically well-defined domain spaces, and (4) to apply the new multi-modal optimization algorithm to uncertainty analysis in the GLUE framework. This dissertation is organized as follows: this chapter introduces the importance of the study, four stand-alone chapters discuss the issues mentioned above, and the last chapter summarizes the findings of the study and draws conclusions.

CHAPTER II

EFFECT OF SPATIAL VARIABILITY IN WATERSHEDS MODELED WITH COMPUTATIONAL TIME STEPS LONGER THAN THEIR TIME OF CONCENTRATION

2.1. Introduction

Many attempts have been made to assess the effect of the spatial variability of hydrologic systems on their response using models (Cotter et al., 2003; Kalin et al., 2003; Chen and Mackay, 2004; Tripathi et al., 2006). The spatial variability of the data used in models includes the spatial discretization of the system into subsystems, data resolution, and the spatial distribution of hydrologic features and parameters. Much has been written about the system discretization and the data resolution (Faurès et al., 1995; Bingner et al., 1997; Manguerra and Engel, 1998; FitzHugh and Mackay, 2000; Andréassian et al., 2001; Di Luzio et al., 2002; Muttiah and Wurbs, 2002; Cotter et al., 2003; Chen and Mackay, 2004; Jha et al., 2004; Chaplot, 2005; Chaplot et al., 2005; Haverkamp et al., 2005; Olivera et al., 2006); however, the effect of the spatial distribution of hydrologic features and parameters is less well known. Although distributed hydrologic models allow the use of spatially distributed information of a watershed, the complexity of the models does not necessarily imply more preferable results (Perrin et al., 2001). A distributed hydrologic model called the Soil and Water Assessment Tool (SWAT) (Arnold et al., 1998) has been used in this chapter to assess the effect of the spatial distribution on the model output.

SWAT was developed to assess the long-term impact of land use and land management changes on hydrologic responses. In SWAT, a watershed is subdivided into subwatersheds, and unique combinations of land use and soil type in each subwater-

shed are referred to as hydrologic response units (HRUs). HRUs are not georeferenced for modeling purposes; that is, their location and spatial distribution within the subwatershed is not taken into account. In fact, spatially disconnected identical combinations of land use and soil type within a subwatershed constitute a single HRU. In SWAT, one stream segment is defined for each subwatershed and is connected dendritically to up and down streams to construct the stream network. Subwatershed size is affected by the threshold value used for stream initiation, which, in turn, affects the stream network density. The stream network, which is the flow routing structure, may play an important role in large watersheds; however, in small watersheds, in which the time of concentration is likely to be shorter than the model computational time step, the routing structure is not expected to be a factor in flow estimation because most raindrops travel through the stream network to the outlet within a single time step.

Studies of the effect of the spatial discretization (i.e., subwatershed size) on flow prediction have been conducted by several researchers. Mamillapalli et al. (1996) found that finer discretization schemes and increased numbers of HRUs improved the runoff flow estimation for a 4,297-km^2 watershed in Texas. However, they also found that there is a threshold beyond which increased model complexity did not lead to better model results, but the model became only computationally more expensive. Haverkamp et al. (2005) demonstrated that model efficiency, according to the Nash-Sutcliffe coefficient (NS) (Nash and Sutcliffe, 1970), improves as more spatial heterogeneity of a watershed is taken into account. They used an entropy function to quantify the heterogeneity of the spatial input data and assumed that the model efficiency is maximized when the entropy of the model parameters becomes equal to that of the spatial modeling units, such as subwatersheds or HRUs. In contrast, for a 21.3-km^2 watershed in northern Mississippi, Bingner et al. (1997) showed that

the prediction of annual runoff volumes was not greatly affected by the number and the size of subwatersheds. They attributed these results to the aggregation effects of subwatersheds and HRUs. Using a 47.3-km^2 watershed in Wisconsin, Chen and Mackay (2004) also found that the level of watershed subdivision does not have a significant influence on annual streamflows. They recommended defining the maximum number of subwatersheds that the model interface allows and one HRU per subwatershed. Mamillapalli (1997), cited in Manguerra and Engel (1998), found that, for eight watersheds ranging in area from 2,000 km^2 to 5,000 km^2, different discretization schemes did not make a significant difference in flow estimation when there were enough HRUs to represent the spatial variability of the watersheds. Manguerra and Engel (1998) also evaluated the effect of discretization schemes on the estimation of monthly streamflow for 3.28-km^2 and 113.38-km^2 watersheds in west central Indiana and a 22.48-km^2 watershed in Mississippi, and showed that the adoption of HRUs with no further spatial subdivision into subwatersheds was sufficient to take into account the spatial variability of the watersheds. They suggested that a detailed discretization method, such as subwatersheds or grid elements, should be applied only when there are site-specific water impoundments or when there is the need for visualizing distributed output. According to FitzHugh and Mackay (2000) and Jha et al. (2004), subwatershed size does not significantly affect monthly and annual streamflow prediction because overall precipitation abstractions are the same regardless of subwatershed size. For 2,000-km^2 and 18,000-km^2 watersheds in Iowa, Jha et al. (2004) found a slight increase in annual streamflow due to increasing transmission gains in subsurface flow and a decrease in transmission losses with decreasing subwatershed size. Olivera et al. (2006) found that simulated daily and monthly flows did not show much difference whether parameter values were assigned to each HRUs based on its hydrologic characteristics or averaged over the subwatersheds. They also noted that

longer times of concentration or shorter computational time steps could affect the estimation of HRU parameters by calibration and, consequently, also flow estimation.

Research on the effect of the resolution of the spatial data (i.e., digital elevation model or DEM, land use, and soil type) on model outputs has shown that model predictions are affected by the aggregation of model parameters. Using six watersheds in Texas ranging in area from 2,807 km^2 to 7,812 km^2, and a 131-km^2 subwatershed, Muttiah and Wurbs (2002) found that runoff flow was not sensitive to the resolution of soil and precipitation data, except for one watershed where wet climate and high soil variability were dominant. Di Luzio et al. (2005) showed that, for a 21.3-km^2 watershed in Mississippi, coarser DEMs resulted in a lower simulated monthly runoff volume due to the reduced accuracy of subwatershed delineation, coarser land use data increased monthly runoff volumes, and coarser soil data decreased monthly runoff volumes. They highlighted that the coarsest DEM data are not adequate for small watersheds because of the inadequate delineation of drainage divides. The DEM resolution has been found to have more impact on model outputs than the land use and soil type data resolutions (Cotter et al., 2003); however, in small watersheds, increasing the DEM resolution does not improve the estimation of mean monthly flows because topography does not affect the estimation of rainfall abstractions by SWAT (Chaplot, 2005).

For a 0.044-km^2 watershed, Faurès et al. (1995) showed that decreasing the number of rain gauges resulted in much uncertainty in runoff prediction. Andréassian et al. (2001) also observed a similar effect when modeling runoff in 71-km^2, 1,120-km^2, and 10,700-km^2 watersheds. However, for a 51-km^2 watershed in central Iowa and a 918-km^2 watershed in central Texas, Chaplot et al. (2005) found that decreasing the rain gauge concentration did not significantly affect the estimation of average monthly runoff flow, which is in contradiction with the previous studies conducted by Faurès

et al. (1995) and Andréassian et al. (2001). Chaplot et al. (2005) attributed this absence of effect of rain gauge concentrations to the averaging effect caused by using monthly model outputs.

In this chapter, the effect on simulated flow values of the spatial distribution of land use, soil type, and precipitation was investigated for a given subwatershed configuration (i.e., stream network). The hypothesis is that, in small watersheds, the spatial variability of land use and soil type does not significantly affect model results because the entire watershed drains within one computational time step, making the location where each raindrop fell immaterial. Various watershed configurations in terms of land use, soil type, and precipitation distribution were considered in this chapter. The model parameters for each watershed configuration was independently calibrated, and spatial, temporal, and spatio-temporal validations were performed to compare model performances. The results of this experiment give insights on the relevance of taking into account the spatial distribution of land use, soil type, and precipitation when modeling small watersheds with a daily computational time step. Section 2.2 describes how to create hypothetical watersheds and how to calibrate the model parameters. Section 2.3 describes the data used in this study and discusses results. Section 2.4 summarizes this chapter and draws conclusions.

2.2. Methodology

In SWAT, runoff from HRUs is calculated either with the Soil Conservation Service (1972) curve number method or the Green and Ampt (1911) infiltration method, aggregated per subwatershed, and routed through the stream network to the watershed outlet using the variable storage routing method (Williams, 1969) or the Muskingum routing method (Lawler, 1964). The stream network can be constructed from the

DEM of the study area or defined based on knowledge of the watershed. The ArcView SWAT interface available in BASINS 3.1 (Di Luzio et al., 2002) was used to develop the SWAT input files for the three watersheds used for this study.

In order to evaluate the effect of the spatial distribution of the model parameters on the modeling results, alternative land use and soil type maps were created as explained below. The National Land Cover Data (NLCD) (USGS, 2006b) was used to create two alternative land use maps. These maps included a single land use map and a randomly distributed land use map. The single land use map had the land use category that was most frequent in the watershed; in the randomly distributed land use map, the total area of each land use was kept, but the location of the cells of each land use were shuffled at random within the watershed. Similarly, the State Soil Geographic database (STATSGO) (USDA-NRCS, 2006) was used to create a single soil type map and a random soil type map. Likewise, because there are multiple rain gauges in the watershed, even with a single land use and soil type, different amounts of runoff are generated in different subwatersheds. To address the effect of the spatial variability of the precipitation, additional models with a single rain gauge were developed for comparison purposes. The single rain gauges were selected based on their proximity to the centroids of the watersheds.

The calibration of the model parameters was performed with the Shuffled Complex Evolution (SCE-UA) algorithm (Duan et al., 1993). The SCE-UA algorithm is widely used in hydrology because it is a robust and efficient global search optimization algorithm (Eckhardt and Arnold, 2001; Muttil and Liong, 2004). In calibration, the spatially distributed parameter values were changed according to a predefined one-parameter rule in order to decrease the number of decision variables and keep the

relative parameter values between spatial modeling units. The rule is

$$p_{\text{new}} = p_0 + \alpha|p_{\text{b}} - p_0| \tag{2.1}$$

where p_{new} is the new parameter value, p_0 is the initial parameter value, p_{b} is either the upper or lower boundary of the parameter value, and α is a real number that varies from -1.0 to 1.0 and is the decision variable evolved by the SCE-UA algorithm. The value of p_{b} is defined as

$$p_{\text{b}} = 0.5p_{\text{u}}(1 + \operatorname{sgn}\alpha) + 0.5p_{\text{l}}(1 - \operatorname{sgn}\alpha) \tag{2.2}$$

where p_{u} and p_{l} are the upper and lower boundaries of the parameter value, respectively, and $\operatorname{sgn}\alpha$ is 1 if α is positive and -1 otherwise. For streamflow calibration, Neitsch et al. (2002b) recommend adjusting the 11 SWAT parameters listed in Table 1. In addition to these parameters, the groundwater delay time (GW_DELAY in the SWAT documentation) and the deep aquifer percolation fraction (RCHRG_DP in the SWAT documentation), which affect groundwater flow and baseflow, were adjusted; that is, the calibration process had 13 decision variables. The objective function used for calibration is the root mean squared error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(Q_i - \hat{Q}_i)^2} \tag{2.3}$$

where Q_i and $\hat{Q}_i$ are the observed and simulated streamflows at the i^{th} day, respectively, and n is the number of days of the calibration period. Note that it is known that the objective function of Eq. (2.3) tends to give more weight to high flows as compared to low flows (Gan and Burges, 1990; Gan and Biftu, 1996; Eckhardt and Arnold, 2001; van Griensven and Bauwens, 2001; Huisman et al., 2003) because errors in high flows are usually greater in absolute value than errors in low flows.

Table 1: List of the model parameters for streamflow calibration. The descriptions were taken from Neitsch et al. (2002b).

Parameter	Description	Range
CN2	Initial NRCS runoff curve number for moisture condition II	35–99
SOL_AWC	Available water capacity of the soil layer (mm H_2O/mm soil)	0.0–1.0
ESCO	Soil evaporation compensation factor	0.01–1.0
GWQMN	Threshold depth of water in the shallow aquifer required for return flow to occur (mm H_2O)	0–5000
GW_REVAP	Groundwater revap coefficient	0.02–0.20
REVAPMN	Threshold depth of water in the shallow aquifer for revap or percolation to the deep aquifer to occur (mm H_2O)	0–500
GW_DELAY	Groundwater delay time (days)	0–200
RCHRG_DP	Deep aquifer percolation fraction	0.0–1.0
CH_K2	Effective hydraulic conductivity in main channel alluvium (mm/hr)	0.025–250
ALPHA_BF	Baseflow alpha factor (days)	0.0–1.0
OV_N	Manning's n value for overland flow	0.01–1.0

After calibration, the models were validated at a different location within the watershed for the calibration period (i.e., spatial validation), at the watershed outlet for a different period (i.e., temporal validation), and at a different location within the watershed and for a different period (i.e., spatio-temporal validation). These validations assessed the applicability under different scenarios of the model parameters obtained by calibration.

The Nash-Sutcliffe coefficient (NS) (Nash and Sutcliffe, 1970) is recommended to assess the performance of hydrologic models (American Society of Civil Engineers, 1993) and defined as

$$\mathrm{NS} = 1 - \frac{\sum_{i=1}^{n}(Q_i - \hat{Q}_i)^2}{\sum_{i=1}^{n}(Q_i - \bar{Q}_i)^2} \tag{2.4}$$

where $\bar{Q}$ is the mean value of the observed flows for the period for which records are available. $\bar{Q}$ is referred to as the "no-model" because no hydrologic concept is used to estimate it. Because the no-model in this study considered all the available historical data, the same no-model was used to calculate the model efficiencies for calibration and validation. Note that the NS coefficient just assesses how a hydrologic model compares to the no-model, and the hydrologic model is expected to outperform the no-model because, otherwise, there would be no justification for its use. In other words, high coefficients do not necessarily imply a good match to the observed values, but could indicate just poor performance of the no-model and vice versa.

Each of the 18 models of each watershed (i.e., each unique combination of land use distribution, soil type distribution, and precipitation distribution) was calibrated independently and, for evaluation purposes, their performance in validation was compared with the performance of the other models.

2.3. Application

2.3.1. Study area and hydrologic data

The three watersheds shown in Figures 1 and 2 were selected for this study because each had a U.S. Geological Survey (USGS) flow gauge at the outlet, another within it, and a period of record of at least 12 years. The areas of the East Fork of the San Jacinto River (08070200), Barton Creek (08155240), and Onion Creek (08159000) watersheds are 1,005 km^2, 227 km^2, and 831 km^2, respectively. A summary of the land use and soil type areas according to the NLCD (USGS, 2006b) and STATSGO (USDA-NRCS, 2006) data is presented in Tables 2 and 3, respectively. Forest is dominant in the East Fork of the San Jacinto River watershed while both forest and rangeland are major land uses in the Barton Creek and Onion Creek watersheds. Sand is the dominant soil type in the East Fork of the San Jacinto River watershed, silt and sand in the Barton Creek watershed, and clay and silt in the Onion Creek watershed. Additionally, the Edwards aquifer's recharge zone underlies across part of the Onion Creek watershed in Hays and Travis counties, with flow gauges 08158700 and 08159000 located upstream and downstream of it, respectively. Likewise, because gauge 08155240 is located upstream of Barton Springs, where a significant discharge from the Edwards aquifer occurs, the Barton Creek watershed is not greatly affected by the Edwards aquifer.

The aquifer's outcrop area is known to be heavily fractured with major faults (Khorzad, 2003; Lindgren et al., 2004). Therefore, the highly permeable soils in the recharge zone were assigned a value of 950 mm/hr for hydraulic conductivity and 0.005 for specific yield as reported by Lindgren et al. (2004) for median values. The other groundwater related parameters were determined in calibration.

The watersheds and subwatersheds were delineated using the 30-meter resolution

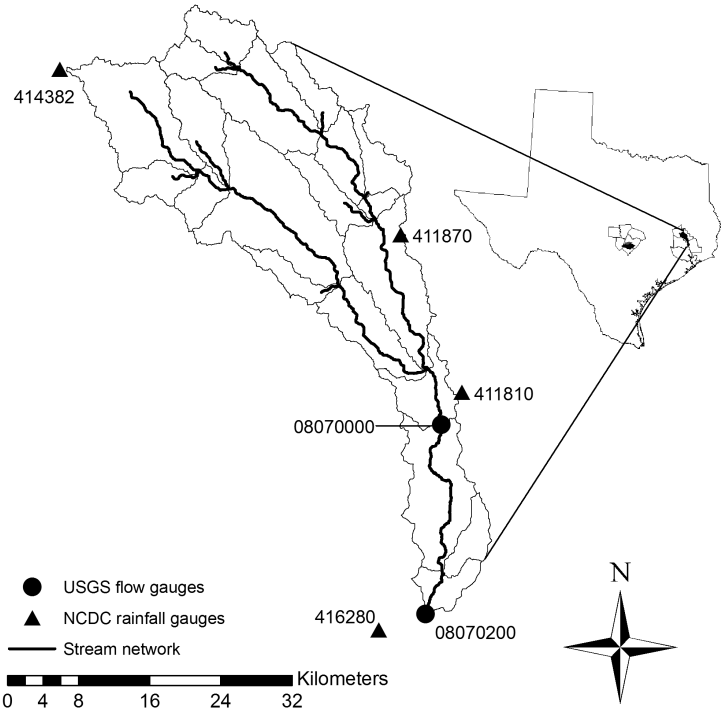


Figure 1: East Fork of the San Jacinto River watershed.

Table 2: Land use distributions according to NLCD 1992.

Watershed	East Fork of the San Jacinto River	Barton Creek	Onion Creek
Urban	2%	6%	9%
Agriculture	1%	1%	3%
Pasture	19%	1%	4%
Forest	72%	51%	45%
Rangeland	4%	40%	38%
Water/Wetland	2%	1%	1%

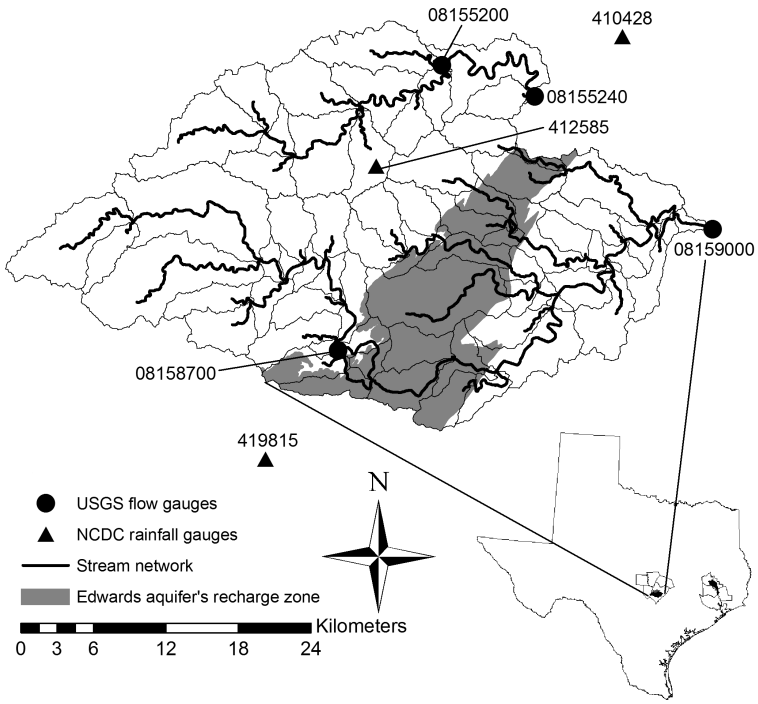


Figure 2: Barton Creek and Onion Creek watersheds.

Table 3: Soil type distributions according to STATSGO. First layer only.

Watershed	East Fork of the San Jacinto River	Barton Creek	Onion Creek
Clay	16%	28%	36%
Silt	22%	37%	35%
Sand	62%	35%	29%

National Elevation Dataset (NED) (USGS, 2006a) with a threshold area of 10 km^2 for the Barton Creek watershed and 20 km^2 for the other two watersheds. The number of subwatersheds delineated in the East Fork of the San Jacinto River, Barton Creek, and Onion Creek watersheds was 20, 16, and 61, respectively. The significantly higher number of subwatersheds in the Onion Creek watershed was caused by the addition of outlet points at the aquifer's boundary.

Daily precipitation and temperature data were obtained from the National Climatic Data Center (NCDC) web site (NOAA-NCDC, 2006). Flow data were obtained from the USGS National Water Information System (NWIS) web site (USGS, 2006c). Rain and flow gauges are shown in Figures 1 and 2. Summaries of the precipitation and streamflow data are presented in Tables 4 and 5, respectively. Multiple rain gauges used in each watershed recorded similar amounts of annual rainfall in both the calibration and validation periods. However, the correlation coefficients between the single rain gauge and the other rain gauges are higher in the East Fork of the San Jacinto River watershed than in the others. Although the amounts of annual rainfall are similar, low correlation coefficients between the rain gauges in the Barton Creek and Onion Creek watersheds imply that their patterns of rising and declining of precipitation depths do not match well. This lack of match suggests that the precipitation fields in these two watersheds were non-uniform while, in the East Fork of the San Jacinto River watershed, it was highly uniform. Therefore, the single gauge models for the East Fork of the San Jacinto River watershed are expected to perform as well as the multiple gauge models while this is not necessarily true for the Barton Creek and Onion Creek models. Table 5 shows that the standard deviations of the daily flows are more than three times greater than the mean values, which is caused by a few number of extremely high flows.

The calibration period ranged from January 1, 1989 to December 31, 1994, and

Table 4: Summary of annual rainfall data. Mean and standard deviation in mm/yr, and Pearson's correlation coefficients against the single rain gauge superscripted by *.

Watershed	NCDC gauge	Calibration period	Validation period
East Fork of the San Jacinto River	411810	1,506 / 228 / 0.74	1,411 / 359 / 0.69
	411870	1,396 / 195 / 1*	1,312 / 358 / 1*
	414382	1,473 / 142 / 0.83	1,272 / 196 / 0.73
	416280	1,546 / 276 / 0.65	1,237 / 520 / 0.69
Barton Creek and Onion Creek	410428	1,053 / 275 / 0.44	933 / 241 / 0.31
	412585	1,051 / 325 / 1*	1,033 / 345 / 1*
Onion Creek	413156	1,038 / 259 / 0.54	979 / 395 / 0.50
	419815	1,089 / 318 / 0.85	1,015 / 478 / 0.79

Table 5: Summary of daily average streamflow. Mean and standard deviation in m^3/s.

Watershed	USGS gauge	Calibration period	Validation period
East Fork of the San Jacinto River	08070200	11.4 / 40.9	9.9 / 35.9
	08070000	10.2 / 40.4	8.4 / 34.3
Barton Creek	08155240	2.2 / 7.7	1.7 / 5.4
	08155200	1.9 / 6.3	1.3 / 4.3
Onion Creek	08159000	3.4 / 20.4	2.9 / 19.9
	08158700	2.1 / 6.4	1.9 / 6.2

the validation period from January 1, 1995 to December 31, 2000. In both cases, the first two years of simulation were used for the initial stabilization of the models, and only four years were used to evaluate the objective function.

2.3.2. Results and discussion

The NS coefficients for calibration and validation are presented in Tables on pages 18 to 23. In calibration, as shown in Table 6, in almost all cases, it was not easy to identify any advantage in choosing one land use or soil type map over the others. Likewise, the multiple rain gauge models performed slightly better than or as well as their corresponding single rain gauge models, except for the case of Onion Creek with random land use and original soil type.

Table 6: Nash-Sutcliffe coefficients in calibration. LO, LS, and LR stand for the land use type of original, single, and random, respectively.

Calibration		Multiple rain gauges			Single rain gauges		
		Soil type			Soil type		
		Original	Single	Random	Original	Single	Random
East Fork of the San Jacinto River	LO	0.44	0.45	0.44	0.42	0.34	0.25
	LS	0.43	0.50	0.46	0.37	0.42	0.41
	LR	0.39	0.37	0.39	0.39	0.37	0.35
Barton Creek	LO	0.86	0.88	0.86	0.83	0.85	0.84
	LS	0.83	0.88	0.88	0.79	0.85	0.85
	LR	0.86	0.87	0.88	0.82	0.84	0.86
Onion Creek	LO	0.92	0.91	0.91	0.92	0.91	0.87
	LS	0.92	0.90	0.91	0.91	0.90	0.89
	LR	0.91	0.91	0.90	0.92	0.90	0.88

The relatively low NS coefficients in the case of the East Fork of the San Jacinto River compared to those of Barton Creek and Onion Creek can be explained by the presence of a few extremely high flows that could not be accurately simulated by

SWAT. There was a severe flood in the East Fork of the San Jacinto River watershed during the calibration period that conveyed 1,320 m^3/s at gauge 08070200 on October 19, 1994, which the model failed to predict. In order to assess the effect of this single value on the calibration results, another set of calibrations were performed excluding this unusually high value. While, in these new models, the sum of the square of the residuals (SSR) significantly decreased, the NS coefficients were not considerably affected because the SSR of the no-model also decreased. For example, for the model with original land use, original soil type, and multiple rain gauges, the SSR of the calibrated models with and without the 1,320 m^3/s flow value were 1,362,433 m^6/s^2 and 384,465 m^6/s^2, respectively, while the SSR of the no-models with and without the high flow value were 2,446,531 m^6/s^2 and 731,300 m^6/s^2, respectively; leading to NS coefficients of 0.44 and 0.47, respectively. That is, the presence of an unusually high flow value and the inability of SWAT to predict it were barely reflected in the NS coefficient.

The calibrated models of the Barton Creek and Onion Creek watersheds successfully predicted high flows; however, their no-models did not, and their SSRs were much greater than those of the calibrated models. Thus, a poor performance of the no-model generated high NS coefficients for these two watersheds.

Table 7 shows the model performance in spatial validation. The East Fork of the San Jacinto River models performed almost as well as in calibration; but the Barton Creek and Onion Creek models did not although their NS coefficients are within the range of acceptable values found in the literature (Hanratty and Stefan, 1998; King et al., 1999; Rosenthal and Hoffman, 1999; Spruill et al., 2000; Eckhardt and Arnold, 2001; Weber et al., 2001; Fontaine et al., 2002; Neitsch et al., 2002c; Eckhardt et al., 2003; Tripathi et al., 2003; Olivera et al., 2006). The reason for the lower performance of the Barton Creek and Onion Creek models might be just the

fact that performances in validation tend to be lower than in calibration or, for the case of Onion Creek, the presence of the Edwards aquifer. For the Onion Creek case, note that flow gauge 08158700 is located right upstream of the Edwards aquifer's recharge zone while flow gauge 08159000 lies on the aquifer's artesian zone. The high rate of groundwater recharge between the two gauges increases uncertainty in the parameter estimates by calibration because less accurate runoff generation from upstream can be compensated for by adjusting groundwater related parameters in the Edwards aquifer.

Table 7: Nash-Sutcliffe coefficients in spatial validation. LO, LS, and LR stand for the land use type of original, single, and random, respectively.

Spatial validation		Multiple rain gauges			Single rain gauges		
		Soil type			Soil type		
		Original	Single	Random	Original	Single	Random
East Fork of the San Jacinto River	LO	0.48	0.38	0.47	0.53	0.36	0.36
	LS	0.47	0.49	0.44	0.47	0.48	0.45
	LR	0.48	0.35	0.46	0.47	0.56	0.49
Barton Creek	LO	0.69	0.72	0.68	0.70	0.75	0.72
	LS	0.65	0.73	0.73	0.67	0.74	0.74
	LR	0.66	0.71	0.73	0.70	0.73	0.75
Onion Creek	LO	0.72	0.76	0.58	0.52	0.55	0.38
	LS	0.70	0.73	0.65	0.45	0.45	0.22
	LR	0.57	0.70	0.43	0.54	0.34	0.46

In the East Fork of the San Jacinto River and Barton Creek models, the single rain gauge models performed as well in spatial validation as the multiple rain gauge models, but were clearly less accurate in the Onion Creek models. In the case of the East Fork of the San Jacinto River, the similarity of the performances in both cases is explained by the fact that the precipitation in the watershed is highly uniform, as indicated by the similar amounts of rainfall and the high correlation between the rain

gauge used in the single rain gauge models (i.e., 411870) and the other rain gauges in the watershed (i.e., 411810, 414382, and 416280) (see Table 4). In the case of Barton Creek, the similarity cannot be explained by uniformity of the precipitation field (see Table 4), but by the fact that the rain gauge used in the single gauge models covers 83% of the watershed area when multiple rain gauges are used. In the case of Onion Creek, the dissimilarity was caused by both non-uniformity of the precipitation field and the fact that the rain gauge used in the single gauge models covers only 58% of the watershed area when multiple rain gauges are used.

Table 8 presents the results of temporal validation. In this case, the multiple rain gauge models of the East Fork of the San Jacinto River performed similar to calibration, but the single rain gauge models performed worse than in calibration, leading to lower NS coefficients. These lower NS coefficients were obtained despite the fact that the precipitation correlation coefficients were somewhat high. In the case of Barton Creek, rain gauge 412585, which was used in the single rain gauge models and covered 83% of the watershed in the multiple rain gauge models, had three days of the precipitation greater than 100 mm (i.e., June 11, 1997, January 7, 1998, and October 18, 1998) that did not generate high flows at the corresponding flow gauges. This mismatch between rainfall and flow caused the NS coefficients to drop below 0, implying that even the no-model performed better than the calibrated models. However, if those three days (out of the 1,461 days of the validation period) are excluded in the calculation of the NS coefficients, their values increase significantly. For example, if the model with original land use, original soil type, and multiple rain gauges is considered, the NS coefficient increases from -0.22 to 0.51, which is lower than in calibration but within the range of NS coefficients documented in the literature for SWAT models (Hanratty and Stefan, 1998; King et al., 1999; Rosenthal and Hoffman, 1999; Spruill et al., 2000; Eckhardt and Arnold, 2001; Weber et al.,

2001; Fontaine et al., 2002; Neitsch et al., 2002c; Eckhardt et al., 2003; Tripathi et al., 2003; Olivera et al., 2006). In the case of Onion Creek, the situation was similar to that in Barton Creek although the NS coefficients did not fall below 0. In this case, rain gauge 412585 was used for the single gauge models and covered 58% of the watershed area in the multiple gauge models. Again, the three days of unusually high precipitation caused large errors which led to low NS coefficients. The models with a single gauge had even lower NS coefficients because of the greater importance of rain gauge 412585 on the overall definition of the precipitation field.

Table 8: Nash-Sutcliffe coefficients in temporal validation. LO, LS, and LR stand for the land use type of original, single, and random, respectively.

Temporal validation		Multiple rain gauges			Single rain gauges		
		Soil type			Soil type		
		Original	Single	Random	Original	Single	Random
East Fork of the San Jacinto River	LO	0.43	0.58	0.50	0.26	0.37	0.25
	LS	0.31	0.62	0.49	0.23	0.47	0.31
	LR	0.33	0.43	0.41	0.30	0.27	0.23
Barton Creek	LO	−0.22	−0.20	−0.22	−0.22	−0.28	−0.01
	LS	−0.44	−0.20	−0.19	−0.39	−0.10	−0.35
	LR	−0.28	−0.32	−0.15	−0.22	−0.10	−0.20
Onion Creek	LO	0.27	0.31	0.31	0.03	0.07	0.04
	LS	0.19	0.27	0.31	0.06	0.05	0.09
	LR	0.21	0.24	0.29	0.12	0.01	0.11

As shown in Table 9, the East Fork of the San Jacinto River models show good performance in spatio-temporal validation while the other two watershed models perform poorly. In the case of Onion Creek, the single rain gauge models performed even better than the multiple rain gauge models. The reason for this bad performance of the multiple gauge models is that, on October 18, 1998, the daily precipitations of 318 mm/day, 58 mm/day, and 37 mm/day were observed at the three rain gauges that

are not used for the single rain gauge models, and of 169 mm/day at the gauge used in the single rain gauge models. Because the recorded value of 318 mm/day affected 34% of the watershed area, significantly higher flows were generated by the multiple rain gauge models, compared to those of the single rain gauge models. For original land use and original soil type, the multiple rain gauge model estimated 306 m^3/s on October 18, 1998 and the single rain gauge model 115 m^3/s while the observed flow was 54 m^3/s. This error causes lower NS coefficients for the multiple rain gauge models than for the single rain gauge ones. When the 318 mm/day precipitation was excluded from the Onion Creek model with original land use, original soil type, and multiple rain gauges, its NS coefficient increased from −0.71 to 0.38.

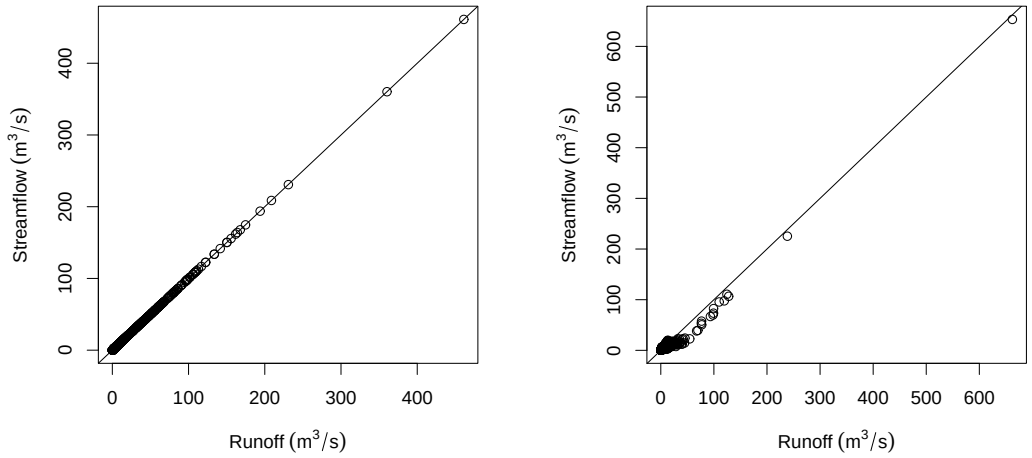
Table 9: Nash-Sutcliffe coefficients in spatio-temporal validation. LO, LS, and LR stand for the land use type of original, single, and random, respectively.

Spatio-temporal validation		Multiple rain gauges			Single rain gauges		
		Soil type			Soil type		
		Original	Single	Random	Original	Single	Random
East Fork of the San Jacinto River	LO	0.56	0.65	0.60	0.43	0.57	0.44
	LS	0.44	0.71	0.57	0.41	0.65	0.52
	LR	0.49	0.58	0.52	0.49	0.48	0.52
Barton Creek	LO	−1.05	−1.00	−1.04	−0.54	−0.61	−0.32
	LS	−1.37	−0.97	−0.95	−0.65	−0.49	−0.69
	LR	−1.15	−1.25	−0.93	−0.53	−0.46	−0.58
Onion Creek	LO	−0.71	−0.48	−0.47	−0.26	−0.28	−0.16
	LS	−0.75	−0.78	−0.02	−0.43	−0.49	−0.40
	LR	−1.21	−1.02	−1.09	−0.11	−0.66	−0.01

Although the NS coefficient is a well-known and commonly used model performance metric, the above results reveal limitations of the coefficient as a measure of model performance. Additionally, because the performance of the no-model highly depends on the characteristics of historical flow records, one should take great caution

when comparing NS coefficients for different watersheds; that is, NS coefficients from different models may be compared only if they have the same no-modal (i.e., same watershed but with different parameter values).

Figure 3 shows plots of the daily-averaged flow at the watershed outlet versus the daily-averaged runoff generated in the watershed, both in m^3/s. Points on the 1:1 line correspond to days in which the flow and the runoff are equal, implying that there are no losses, and the draining time is less than one day. On the contrary, points off the 1:1 line reflect losses or residence times in the watershed longer than one day. For the East Fork of the San Jacinto River and Barton Creek models, almost all of the points are found on the 1:1 line (see Figure 3a); for the Onion Creek models, flows are lower than runoff because part of the runoff is lost to the Edwards aquifer's recharge zone before reaching the watershed outlet (see Figure 3b).



(a) East Fork of the San Jacinto River (b) Onion Creek

Figure 3: Streamflow versus runoff for selected models out of the 54 calibrated models. (a) The East Fork of the San Jacinto River model with original land use, original soil type, and multiple rain gauges. (b) The Onion Creek model with single land use, single soil type, and single rain gauge.

The Manning's channel roughness coefficient (CH_N2 in the SWAT documen-

tation) affects flow velocities and travel times (Hsu et al., 2006), and, consequently, residence times in a watershed. To assess the effect of the Manning's channel roughness coefficient on watershed responses, the calibrated models were run with the Manning's coefficients of 0.001 and 0.01 to 0.10 at 0.01 intervals. It was observed that the effect of the different Manning's coefficient values on the hydrographs at the watershed outlets was negligible because, in all cases, the time taken in flow routing was shorter than one day. This impact of time resolution on the results of small watershed models is conceptually similar to the buffering effect that reduces the importance of daily rainfall variability by using monthly flow estimates (Chaplot et al., 2005).

However, the most important observation is that, despite the expectation that the original spatial distribution of land use and soil type would yield better predictions in spatial validation, the random and lumped models performed overall as well as the original ones. In other words, the location within the watershed of specific land use and soil type areas did not significantly affect the performance of the models. The reason for the consistency in the performance level, in spite of the change in land use and soil type distribution, is caused by the fact that the entire watershed drains within a single computational time step making it irrelevant to identify where runoff was generated. However, it was noticed that multiple rain gauge models performed better than single rain gauge models, both in temporal and spatial validation. In fact, for the specific dataset used in this study, it was found that accounting for the spatial variability of rainfall—as a means for estimating the precipitation volume more accurately—was more important than accounting for the spatial variability of land use and soil type. Although the three watersheds analyzed in this study presented different hydrologic characteristics, generalization of these conclusions should be done carefully.

2.4. Summary and conclusions

This chapter discussed the impact of modifying the spatial distribution of land use, soil type, and precipitation on estimated streamflows in watersheds with times of concentration shorter than the model computational time step. The SWAT model was used to estimate the hydrographs at the watershed outlets. Different representations of the spatial data resulted in comparable model performances, and even the use of uniform land use and soil type maps was not noticeable. In general, spatially distributed data help understand the characteristics of a watershed and provide distributed hydrologic models valuable information; however, when the watershed size is small compared to the time step of the model, realistic representations of the spatial data do not necessarily improve the model performance. When the model computational time step is long compared to the watershed time of concentration, the model can only capture the lumped effect of the hydrologic processes at the watershed outlet. Despite the diversity of the hydrologic characteristics of the watersheds analyzed in this chapter, generalization of these conclusions presented here should be done carefully. In fact, watersheds in which a correlation between land use or soil type, and precipitation distribution exists might require accounting for the spatial variability for estimating more accurately the runoff volume.

CHAPTER III

ENHANCED SPECIATION IN PARTICLE SWARM OPTIMIZATION FOR MULTI-MODAL PROBLEMS

3.1. Introduction

Due to the complexity and the non-linear nature of real-world optimization problems, it is often not possible to analytically derive the derivative of the objective function (Duan et al., 1994; Xiong and O'Connor, 2000; Sotiropoulos et al., 2002; Zhang et al., 2004). The lack of analytical derivatives of the objective function forces the use of direct search approaches, which only require function evaluations (Xiong and O'Connor, 2000; Sotiropoulos et al., 2002; Brath et al., 2004; Zhang et al., 2004; Wei and Zhao, 2005; Yang et al., 2005). However, even direct search algorithms, such as the simplex method, can suffer from early convergence to local optima or slow convergence (Trabia, 2004). In order to overcome these shortcomings, population-based search algorithms have received much attention to improve the exploration of the search space (Sotiropoulos et al., 2002).

Holland (1962) formalized the analogy between optimization problems and the natural selection of organisms leading to evolutionary algorithms. Random searches based on artificial genetic evolution are referred to as Genetic Algorithms (GAs) (Guanqi and Shouyi, 2003). In GAs, a generation of artificial organisms evolve through genetic operators such as crossover, mutation, and random selection (Chelouah and Siarry, 2003). Inspired by the social behavior of ant colonies, Dorigo et al. (1996) introduced the Ant System (AS) metaheuristic, which is the predecessor of the Ant Colony Optimization (ACO) algorithm. In ACO's basic algorithm (Dorigo et al., 1996), artificial ants initially move randomly in the stratified search space to

find solutions and lay down pheromone trails representing the probability of possible solutions being the global optimum while other artificial ants probabilistically prefer to follow pheromone deposits rather than walk randomly to find the shortest path to solutions. Pheromone deposited on non-optimal routes tends to evaporate as the number of artificial ants following the routes decreases to reduce the probability of the routes being optimal. The collective behavior of these simple artificial ants eventually finds the shortest path to the global optimum. Since ACO was originally developed for combinatorial optimization, several attempts have been made to design continuous pheromone models to solve continuous problems (Socha and Dorigo, 2006).

Another promising optimization method based on collective artificial intelligence is Particle Swarm Optimization (PSO), introduced by Eberhart and Kennedy (1995) and Kennedy and Eberhart (1995). PSO is a metaheuristic approach mimicking the movement of birds in a flock sharing information with each other (Acan and Gunay, 2005). Individuals in the swarm represent parameter samples referred to as particles. Each particle in the swarm keeps track of its own best solution achieved so far and shares the information with topological neighbors to fly toward an optimal solution (Brits et al., 2007). The challenge in high-dimensional problems has been to locate the global optimum without trapping into local optima (Hendtlass, 2003). Different topological neighbors have been studied to enhance the success rate of finding the global optimum (Brits et al., 2007). These topologies include, among others, the gbest (i.e., global best) method, the lbest (i.e., local best) method (Eberhart et al., 1996), the von Neumann topology (Kennedy and Mendes, 2002), and the spatial topology (Suganthan, 1999). However, there are usually more than one possible solution to real-world problems (Hendtlass, 2003; Brits et al., 2007) (e.g., multi-objective problems and multi-modal problems).

Multi-objective optimization seeks to find the non-inferior solutions referred to

as the Pareto-optimal set (Gupta et al., 1998; Yapo et al., 1998; Madsen, 2003). In the context of multi-objective optimization, one has to choose one or more trade-off solutions from conflicting scenarios distributed along the Pareto-optimal front (Deb, 2004). However, because not all objective functions can be mathematically formulated or considered in optimization, trade-off solutions can turn out to be not on the Pareto-optimal front when additional objective functions are taken into account (Brill, 1979). Even if the non-inferior solutions are chosen for decision making, there is a chance that these solutions are not realistically or even mathematically the best (Brill, 1979; Zechman and Ranjithan, 2003, 2007).

It is often useful to obtain "maximally different" sub-optimal alternative solutions in order to gain insight to feasible solutions (Brill, 1979; Zechman and Ranjithan, 2007). In multi-modal problems, there may be more than one solution if global and local optima are considered (Li, 2004). However, unlike in multi-objective problems, there is no Pareto-optimal front because there is only one objective function in multi-modal problems (Brits et al., 2007). However, strict criteria to determine whether or not local optima were found cannot be well defined (Brits et al., 2007) because the surface of the objective function is not known a priori, and local optima have mathematically inferior fitness values compared to the global one.

PSO's adaptive control of particles' movement and of their topological neighbors improves the exploration of the search space, which is a promising feature for multi-modal problems (Li, 2004). Cluster analysis techniques have been employed to enhance the exploration of the search space (among others, Kennedy, 2000; Li, 2004). Kennedy (2000) applied k-means algorithms to classify particles into groups, which individually converge to their own optimum. However, the main pitfall of this method is that the number of clusters must be specified in advance without knowing how many optima actually exist (Li, 2004). The Species-based PSO (SPSO), which

uses proximity-based grouping called "speciation," was introduced by Li (2004) to alleviate the shortcomings of k-mean classifications. Because the speciation radius is used to classify the swarm population into subpopulations, referred to as "species," there is no need to pre-specify the number of clusters (Li, 2004). Likewise, NichePSO (Brits et al., 2002) creates subswarms, each of which employs the Guaranteed Convergence PSO (GCPSO) (van den Bergh and Engelbrecht, 2002) to ensure convergence to different local optima. GCPSO adaptively updates the velocity of the best particle based on its number of consecutive successes or failures to avoid premature convergence (van den Bergh and Engelbrecht, 2002). Parsopoulos and Vrabatis (2001) applied function stretching (Parsopoulos et al., 2001) to transform the objective function right after the swarm finds an optimum so that particles approaching already found optima are assigned a low fitness value to prevent them from revisiting the previous solutions. The transformation of original objective functions is known to bring in new local optima (van den Bergh, 2002).

Diversity of particles is a measure to how well particles are distributed in the search space, and it is necessary to maintain high diversity during the optimization process in order to escape from local optima (Riget and Vesterstrøm, 2002; Monson and Seppi, 2006) or find them dispersed in the search space. Unlike in previous implementations of PSOs, particles in the Spatial Extension PSO (SEPSO) (Krink et al., 2002) have volumes, and there is a collision detection algorithm so that particles do not overlap with each other, thus, avoiding clustering around local optima. This method also prevents unnecessary function evaluations when particles are close to each other (Monson and Seppi, 2006). The Attractive and Repulsive PSO (ARPSO) (Riget and Vesterstrøm, 2002) artificially keeps diversity within the prespecified range through "attraction" and "repulsion." ARPSO measures the diversity of a swarm by the average distance of particles from the centroid of the swarm. Mon-

son and Seppi (2006) extended SEPSO in the Contracting Radius, Increasing Bounce SEPSO (CRIBS) to adaptively update the radius and distance of collision detection.

In this chapter, SPSO was extended not only to find multiple optima but also to better explore the search space. Enhanced exploration will increase the probability of finding more solutions, which is a preferable feature for multi-modal optimization. Section 3.2 describes the particle's movement in PSO and discusses the extensions to SPSO introduced in this study. Sections 3.3 and 3.4 discuss the experimental setup and results, respectively. Section 3.5 summarizes this chapter and draws conclusions.

3.2. Isolated Speciation-based PSO

In this study, SPSO was modified to enhance the discovery of sub-optimal yet potentially attractive solutions. In addition, a non-random deterministic sampling strategy was employed to increase the uniformity of particle samples. In the Isolated Speciation-based PSO (ISPSO), possible solutions found by the swarm are called "nests," and finding them is referred to as "nesting." The ISPSO algorithm is given in Figure 4.

3.2.1. Particle's movement

In PSO, each particle represents a parameter sample, and the swarm consists of a population of particles. Particles in the swarm share their information with topological neighbors to move around the search space toward optimal solutions. In a D-dimensional problem space, particle i's current position and velocity are $\vec{x}_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ and $\vec{v}_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, respectively. Particle i uses its own best position $\vec{p}_i$ (i.e., private best or pbest) and the global best position $\vec{p}_g$ (i.e., gbest) that the swarm has achieved so far to determine its velocity at the next time step.

Define a problem: D, $\vec{x}_{\min}$, $\vec{x}_{\max}$, $\vec{v}_{\max}$, $f(\vec{x})$
Initialize the ISPSO parameters: $|S|$, r_{species}, r_{prey}, r_{nest}
Initial population from the scrambled Sobol' sequence
$N \leftarrow \emptyset$: Storage for nests
repeat
  Evaluate $f(\vec{x}_i)$ and $\vec{p}_i$ for $i = 1, \ldots, |S|$.
  Increase the ages of all particles by 1.
  $A \leftarrow S$ sorted in the decreasing order of fitness
  $B \leftarrow \emptyset$: Species seeds
  $C \leftarrow \emptyset$: Particles participating in speciation
  for all $\vec{a} \in A$ **do** {Proximity-based speciation}
    found $\leftarrow$ FALSE
    for all $\vec{b} \in B$ **do**
      if $d(\vec{a}, \vec{b}) \leq r_{\text{species}}$ **then**
        found $\leftarrow$ TRUE
        $C \leftarrow C \cup \{\vec{a}, \vec{b}\}$
        $\vec{p}_i \leftarrow \vec{b}$: i indicates the unsorted index of $\vec{x}_i = \vec{a}$.
        break
    if found = FALSE **then**
      $B \leftarrow B \cup \{\vec{a}\}$
      $\vec{p}_i \leftarrow \vec{a}$: i indicates the unsorted index of $\vec{x}_i = \vec{a}$.
      Modified speciation according to Figure 5
  Isolated speciation according to Figure 6
  Update velocity $\vec{v}_i$ for $i = 1, \ldots, |S|$.
  for all $\vec{n} \in N$ **do** {Turbulence of species seeds}
    for all $\vec{b} \in B$ **do**
      if $d(\vec{b}, \vec{n}) < 2 \cdot r_{\text{nest}}$ **then**
        Add a small turbulence to the velocity of $\vec{b}$.
  Check for convergence, and add solutions to N if any.
  Update position $\vec{x}_i$ for $i = 1, \ldots, |S|$.
  Fitness assimilation according to Eqs. (3.12) and (3.13).
  Preemptive nesting
until The maximum number of iterations is reached.

Figure 4: ISPSO algorithm.

The d^{th} dimensional element of particle i's velocity at time step $t+1$ is updated as

$$v_{i,d}(t+1) = v_{i,d}(t) + \psi_1 r_1(t)(p_{i,d} - x_{i,d}) + \psi_2 r_2(t)(p_{g,d} - x_{i,d}) \tag{3.1}$$

where ψ_1 and ψ_2 are the cognitive and social coefficients, respectively, that are set as control parameters prior to an optimization run, and $r_1(t)$ and $r_2(t)$ are random numbers from the uniform distribution between 0 and 1. Different topological neighbors use different $\vec{p}_g$ definitions referred to as the local best $\vec{p}_l$ (i.e., lbest), which is the best position obtained by an individual neighbor, not by the entire swarm.

Shi and Eberhart (1998) introduced the inertia weight ω to stabilize particle movement by modifying Eq. (3.1) as follows:

$$v_{i,d}(t+1) = \omega v_{i,d}(t) + \psi_1 r_1(t)(p_{i,d} - x_{i,d}) + \psi_2 r_2(t)(p_{g,d} - x_{i,d}) \tag{3.2}$$

where ω is the inertia weight typically decreasing from 0.9 to 0.4 during optimization (Eberhart and Shi, 2000). Clerc (1999), in turn, proposed the constriction factor χ to efficiently control particle velocities and extended Eq. (3.1) as

$$v_{i,d}(t+1) = \chi(v_{i,d}(t) + \psi_1 r_1(t)(p_{i,d} - x_{id}) + \psi_2 r_2(t)(p_{g,d} - x_{i,d})) \tag{3.3}$$

where χ is defined as

$$\chi = \frac{2}{|2 - \psi - \sqrt{\psi^2 - 4\psi}|} \tag{3.4}$$

where $\psi = \psi_1 + \psi_2$ and $\psi > 4$. The constriction factor is known to outperform the inertia weight in improving the stability of convergence by preventing the explosion of particle velocities (Eberhart and Shi, 2000; Blackwell and Branke, 2006) and is used in ISPSO.

The particle velocity is constrained between $-\vec{v}_{\max}$ and $\vec{v}_{\max}$. Particle i's position

at time step $t+1$ is updated as

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t+1) \tag{3.5}$$

The pre-specified range $[\vec{x}_{\min}, \vec{x}_{\max}]$ is used to restrict $\vec{x}_i(t+1)$ to the feasible search space.

3.2.2. Sampling strategy for high diversity

In ISPSO, to ensure uniformity rather than randomness of the particle distribution in the search space, swarm particles are initialized with low-discrepancy sequences or quasi-random sequences instead of pseudo-random sequences (i.e., uniform distribution). The uniformity of the point set $\vec{x}_1, \ldots, \vec{x}_n \in I = [0,1)^s$ is measured by the star discrepancy:

$$D_n^*(X) = \sup_{A \in J^*} \left| \frac{N(A,X)}{n} - \lambda_s(A) \right| \tag{3.6}$$

where $X = \{\vec{x}_1, \ldots, \vec{x}_n\}$, J^* is the set of subintervals of I of the form $\prod_{i=1}^{s}[0, u_i)$, $0 \leq u_i \leq 1$, n is the number of points in X, $N(A,X)$ is the number of points in X belonging to an s-dimensional hyperrectangle $A = \prod_{i=1}^{s}[0, u_i)$, and $\lambda_s(A)$ is the Lebesgue measure of the hyperrectangle A. The Lebesgue measure on A is defined as $\lambda_s(A) = \prod_{i=1}^{s} u_i$. Since the discrepancy measures how much samples are deviated from the spatial uniformity of the particle distribution, a low discrepancy value indicates a high uniformity in the particle distribution. Galanti and Jung (1997) compared three low-discrepancy sequences including the Sobol' (Sobol', 1967), Halton (Halton, 1960; Halton and Smith, 1964), and Faure (Faure, 1982) sequences. They concluded that the Sobol' sequence is superior to the others in terms of speed and degradation of low discrepancy in high-dimensional spaces (e.g., even in the 260^{th} dimension).

Antonov and Saleev's (1979) improvement to the Sobol' sequence is used in ISPSO. Suppose $p^j(x)$ is a primitive polynomial of degree q and of modulo 2 for the j^{th} dimension as defined as

$$p^j(x) = x^q + a_1 x^{q-1} + \cdots + a_{q-1}x + 1 \tag{3.7}$$

where $a_1, \ldots, a_{q-1}$ arbitrarily take either 0 or 1. $p^j(x)$ is said to be primitive if the order of $p^j(x)$ is $2^q - 1$. The order of the polynomial is the smallest integer e such that $x^e + 1$ can be divided by $p^j(x)$. For example, the order of $f(x) = x^2 + x + 1$ is $3 = 2^2 - 1$, and $f(x)$ is primitive because $(x^2 + x + 1)(x + 1) = x^3 + 2x^2 + 2x + 1 = x^3 + 0x^2 + 0x + 1 = x^3 + 1$. However, $g(x) = x^2 + 1$ has an order of $4 \neq 2^2 - 1$ because $(x^2 + 1)(x + 1) = x^3 + x^2 + x + 1 \neq x^3 + 1$ but $(x^2 + 1)(x^2 + 1) = x^4 + 2x^2 + 1 = x^4 + 0x^2 + 1 = x^4 + 1$. Note that $p^j(x)$ is a polynomial of modulo 2. Different $p^j(x)$'s are used for different dimensions. Now, the q-term recurrence relation M_i^j is defined as

$$M_i^j = 2a_1 M_{i-1}^j \oplus 2^2 a_2 M_{i-2}^j \oplus \cdots \oplus 2^{q-1} a_{q-1} M_{i-q+1}^j \oplus 2^q M_{i-q}^j \oplus M_{i-q} \tag{3.8}$$

where $q < i \leq w$, and $\oplus$ is the bitwise exclusive disjunction operator (XOR) defined as follows: $0 \oplus 0 = 1 \oplus 1 \equiv 0$ and $0 \oplus 1 = 1 \oplus 0 \equiv 1$. M_i^j for $i = 1, \ldots, q$ is arbitrarily determined such that M_i^j is an odd integer satisfying $0 < M_i^j < 2^i$. A set of binary fractions called the direction number is defined as

$$v_i^j = M_i^j \cdot 2^{-i} \tag{3.9}$$

where $1 \leq i \leq w$. The Sobol' number x_{n+1}^j for $0 \leq n < 2^w - 1$ is generated as follows:

$$x_{n+1}^j = x_n^j \oplus v_c^j \tag{3.10}$$

where $x_0^j = 0$, and c is the rightmost zero bit of n. The maximum allowed number of sequence generations for each dimension is $2^w - 1$. See Galanti and Jung (1997) for more details including examples.

Owen (1998) and Faure and Tezuka (2002) successfully applied random scrambling schemes to the Sobol' sequence to increase uniformity. In ISPSO, particle generation is controlled by the Sobol' sequence with both the Owen and Faure-Tezuka scrambling techniques.

3.2.3. Isolated speciation

SPSO groups particles in the swarm into species based on proximity. Proximity is measured by the Euclidean distance d between particles i and j as follows:

$$d(\vec{x}_i, \vec{x}_j) = \sqrt{\sum_{d=1}^{D} (x_{id} - x_{jd})^2} \tag{3.11}$$

where $\vec{x}_i$ and $\vec{x}_j$ are the positions of particles i and j, respectively, and D is the problem dimension.

Speciation takes place at every iteration of the optimization process because the movement of particles in the search space continuously changes the distance between particles. The best particle in a species is referred to as the species seed and is considered the local best in the species (i.e., $\vec{p}_g$ in Eq. (3.3)). When any species converges to a certain point, particles belonging to the species would not move actively because the species seed's local and individual best positions (i.e., $\vec{p}_g$ and $\vec{p}_i$ in Eq. (3.3), respectively) will be very close to each other. Because this stagnation can cause premature nesting, in ISPSO, the memory of the current particle generation is used to refine the local best of any newly created species seed at each iteration. When a species is generated, the seed checks if there are particles with better fitness values than its

own fitness within the species radius r_{species} as shown in Figure 5. This may happen when there are superior particles within the region of the species being generated that belong to the superior species. In this case, the local best of the new species seed is set to the current position of the superior particle. Furthermore, the seed also checks the individual best position of each particle in the swarm to see if any better positions were explored in the past near the region of the species being generated. If the seed found better positions than itself, the local best of the seed is set to the best position found, and the seed and particles in the species may not stay around the seed's current position. In this way, speciation learns about the region of a new species from the memories of all the particles in the swarm.

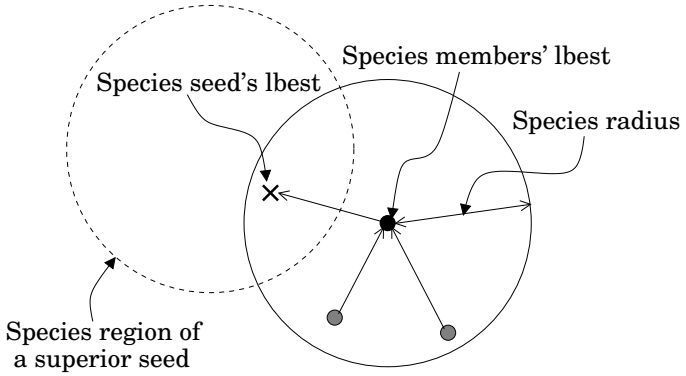


Figure 5: Modified speciation. The black dot indicates the seed of a new species being generated, the gray dots are the members of the new species, and the "X" symbol is either a member of an overlapping species or the private best (i.e., pbest) of a particle in the current population that has a better fitness value than the new species seed. Lbest denotes the local best of a particle.

Preliminary experiments in this study showed that SPSO requires more particles compared to other PSOs in order to achieve convergence. Particles in SPSO need to be close enough to form species while they have to create multiple species at the same time for exploration. Otherwise, there can be isolated particles that do not participate in any speciation. The threshold radius for speciation r_{species} needs to be

kept small to ensure finding the solutions that are close to each other. Since each species seed represents a candidate nest, the species radius is critical for multi-modal optimization. The question arises of how many particles are required to make sure enough species are formed. If particles are too sparsely distributed, isolated particles are not able to move when their initial velocities are set to 0. Even if initial velocities are set to non-zero values, there is still a chance of generating isolated particles, which only consume function evaluations at fixed positions without any contribution to finding solutions. In order to overcome this shortcoming, as shown in Figure 6, isolated particles form an additional species in ISPSO, and their ages (i.e., the number of consecutive iterations during which a particle has participated in speciation) are reset to one because their experiences should not be trusted as much as those of the other particles. By incorporating this strategy, the swarm size is significantly decreased to the same level as other PSOs.

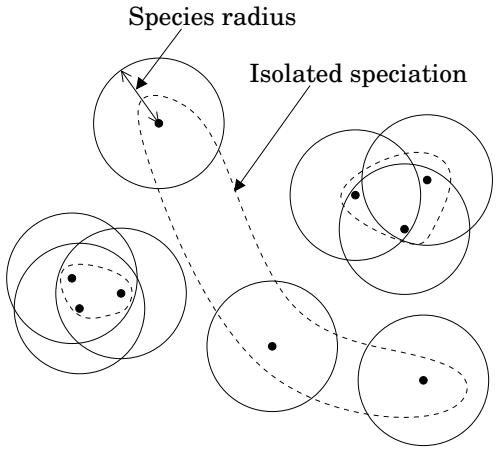


Figure 6: Isolated speciation. The black dots are particles, the solid circles are their speciation regions, and the particles in the dashed splines generate species.

3.2.4. Fitness assimilation

SEPSO introduces the concept of particle volumes (Krink et al., 2002). The idea is that a collision between volumetric particles helps prevent clustering which attracts particles to a few sub-optimal solutions. ISPSO also adopts this concept through the pre-specified radius r_{prey}, but a collision does not make particles bounce from each other. Instead of bouncing, fitness assimilation or preying takes place between two particles in contact. That is, the past experiences of the two particles are combined to create a new particle. Each particle keeps three memories including its current position, current velocity, and individual best position. The current position of the new particle is based on the fitness values at the current positions of the two particles. The position and velocity of the particle with the higher fitness are assigned to the new particle. Similarly, the new particle's individual best position is assigned the better individual best position of the two original particles. The new particle replaces one of the two original particles. If f denotes the objective function for minimization problems, the formulation of new particle generation is as follows:

$$(\vec{x}_{\text{new}}, \vec{v}_{\text{new}}) = \begin{cases} (\vec{x}_i, \vec{v}_i) & \text{if } f(\vec{x}_i) \leq f(\vec{x}_j), \\ (\vec{x}_j, \vec{v}_j) & \text{otherwise;} \end{cases} \tag{3.12}$$

$$\vec{p}_{\text{new}} = \begin{cases} \vec{p}_i & \text{if } f(\vec{p}_i) \leq f(\vec{p}_j), \\ \vec{p}_j & \text{otherwise} \end{cases} \tag{3.13}$$

where $\vec{x}_{\text{new}}$, $\vec{v}_{\text{new}}$, and $\vec{p}_{\text{new}}$ are the position, velocity, and individual best position of the new particle, respectively, $\vec{x}_i$ and $\vec{x}_j$ are the particles' original positions, $\vec{v}_i$ and $\vec{v}_j$ are the particles' original velocities, and $\vec{p}_i$ and $\vec{p}_j$ are the particles' original individual best positions. To replace the remaining particle, one additional particle is quasi-randomly generated using the scrambled Sobol' sequence described previously.

This new particle fills a gap between previously quasi-randomly generated particles.

3.2.5. Nesting criteria for global and local optima

Because there are no absolute stopping criteria when searching for local optima (Brits et al., 2007), it is difficult to claim that a local optimum has been found. This lack of stopping criteria also applies to searching for the global optimum when the fitness value is not well defined. Ho et al. (2005) and Ho et al. (2006) introduced the age of a particle to enhance the exploration of the search space. In ISPSO, the pre-specified age threshold a is used to determine whether the experience of an old particle should be trusted or not. If the age of any species seed exceeds the age threshold, and the normalized geometric mean of its positions and the standard deviation of its fitness values during the most recent 50% of lifetime are small enough, the seed is considered a possible solution referred to as a nest.

The standard deviation of the fitness values of particle i from age $\lfloor a_i/2 + 0.5 \rfloor$ to a_i is used for nesting. $\lfloor \cdot \rfloor$ and a_i denote the flooring function and the current age of particle i, respectively. The threshold value for the standard deviation is called ϵ_f. However, a small standard deviation of the fitness value does not guarantee the convergence of a particle when the particle hops around points with very similar fitness values (e.g., a near-plateau region).

Liong and Muttil (2001) used the normalized geometric mean as the stopping criterion for the Shuffled Complex Evolution (SCE-UA) algorithm developed by Duan et al. (1992). In order to attain increased accuracy in nesting, ISPSO also uses the normalized geometric mean defined as follows:

$$\left(\prod_{j=1}^{D} \frac{x_{i,j}^{+} - x_{i,j}^{-}}{x_{\max,j} - x_{\min,j}} \right)^{1/D} = \exp\left(\frac{1}{D} \sum_{j=1}^{D} \log \frac{x_{i,j}^{+} - x_{i,j}^{-}}{x_{\max,j} - x_{\min,j}} \right) \tag{3.14}$$

where $x_{i,j}^{+}$ is $\max\{x_{i,j}^{k} \mid \lfloor a_i/2 + 0.5 \rfloor \leq k \leq a_i\}$, $x_{i,j}^{-}$ is $\min\{x_{i,j}^{k} \mid \lfloor a_i/2 + 0.5 \rfloor \leq k \leq a_i\}$, and $x_{\max,j}$ and $x_{\min,j}$ are the j^{th} dimensional upper and lower bounds of the search space, respectively. $x_{i,j}^{k}$ denotes the j^{th} dimensional position of particle i at age k. A small value of the normalized geometric mean implies that particle i has converged. The threshold value for the normalized geometric mean is called ϵ_x.

3.2.6. Preemptive nesting

After having found a number of solutions, the swarm should avoid converging to the existing nests to prevent unnecessary function evaluations at positions near the known possible solutions. This is accomplished by substituting particles falling within the nesting radius r_{nest} of the existing solutions with new quasi-random particles generated from the scrambled Sobol' sequence. As a side effect of this behavior, a flat circular area is created that has the same objective function value as the solution within the nesting area, causing some particles to cluster around the edge of the region. In order to prevent convergence near the edge of this area, a "turbulence region" surrounding the nesting area is defined so that any species seed moving into this region is assigned a small random velocity, and individuals belonging to this species may not be able to converge.

3.3. Experiments

ISPSO was tested on the five functions suggested by Beasley et al. (1993). These test functions were modified to be suitable for minimization problems as follows:

$$F1(x) = 1 - \sin^6(5\pi x), \tag{3.15}$$

$$F2(x) = 1 - \exp\left(-2\log(2) \cdot \left(\frac{x - 0.1}{0.8}\right)^2\right) \cdot \sin^6(5\pi x), \tag{3.16}$$

$$F3(x) = 1 - \sin^6\left(5\pi(x^{3/4} - 0.05)\right), \tag{3.17}$$

$$F4(x) = 1 - \exp\left(-2\log(2) \cdot \left(\frac{x - 0.08}{0.854}\right)^2\right) \cdot \sin^6\left(5\pi(x^{3/4} - 0.05)\right), \tag{3.18}$$

$$F5(x_1, x_2) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2. \tag{3.19}$$

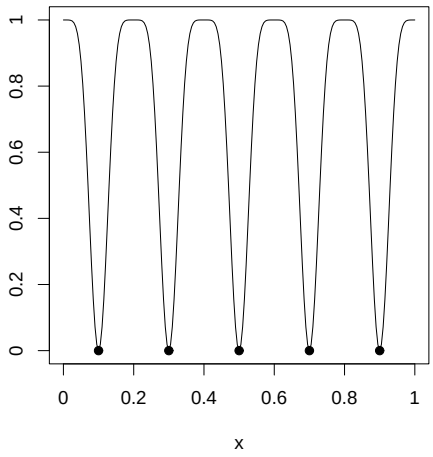
Figure 7 shows the minima of the five functions. The search spaces are $[0, 1]$ for $F1$ to $F4$ and $[-6, 6]^2$ for $F5$, the same search spaces used in Beasley et al. (1993). $F1$ has equally spaced minima at 0.1, 0.3, 0.5, 0.7, and 0.9; $F2$ has almost equally spaced minima at 0.100, 0.299, 0.499, 0.698, and 0.898; $F3$ has minima at 0.080, 0.247, 0.451, 0.681, and 0.934; and $F4$ has minima at 0.080, 0.246, 0.449, 0.679, and 0.930. $F1$ and $F3$ have five global minima while $F2$ and $F4$ have one global minimum and four local minima. $F5$ has four almost equal minima at $(3.58, -1.86)$, $(3.0, 2.0)$, $(-2.815, 3.125)$, and $(-3.78, -3.28)$.

Additionally, two scalable massively multi-modal functions (i.e., the Rastrigin and Griewank functions) were used to test the scalability of ISPSO. The Rastrigin function is defined as

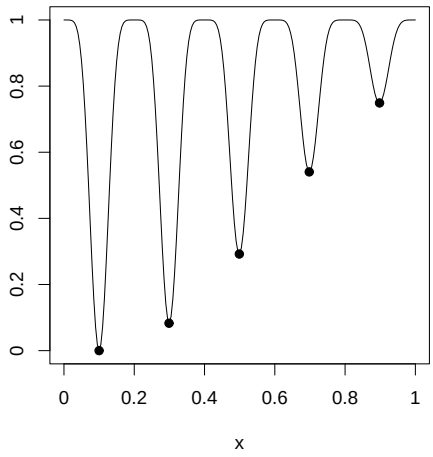
$$F6(\vec{x}) = \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right] \tag{3.20}$$

and the Griewank function is defined as

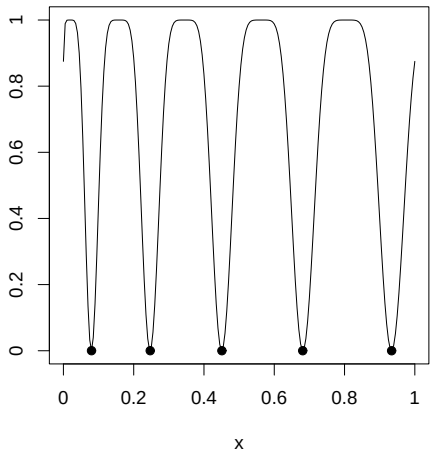
$$F7(\vec{x}) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{3.21}$$



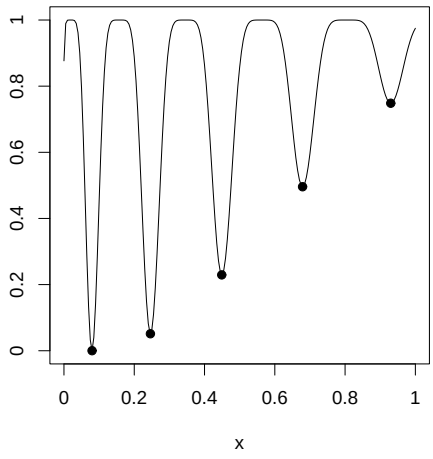
(a) $F1$



(b) $F2$

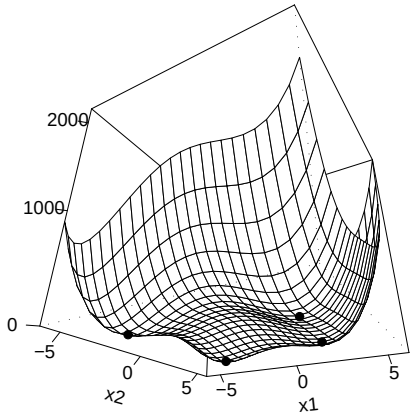


(c) $F3$



(d) $F4$

Figure 7: Test functions for performance test. The black dots indicate the locations of global and local minima.

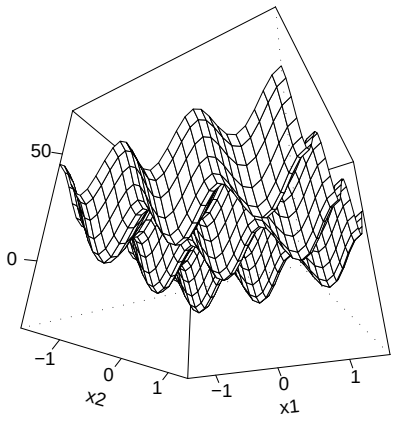


(e) $F5$

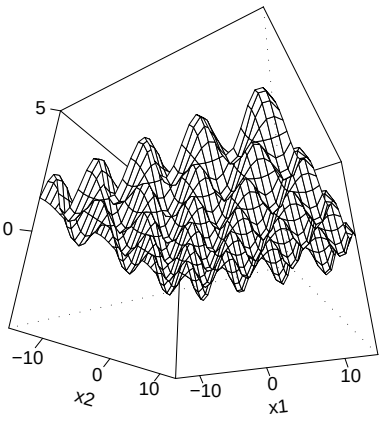
Figure 7: Continued.

where D is the dimension of the function. Figure 8 shows the surfaces of the two functions. The search space was restricted to $[-1.5, 1.5]^D$ for $F6$ and $[-14, 14]^D$ for $F7$.

Because it is difficult, if not impossible, to well define the fitness values of global and local minima in real-world applications, optimization algorithms are not able to use those fitness values in their stopping criterion. Although, in all test runs, ISPSO was tested on mathematical functions with known minima and minima fitness values, it was assumed that ISPSO did not know the fitness values of global and local minima; that is, the algorithm was not supposed to know errors between the true solutions and nests during optimization, and no threshold values for the errors were used to find possible solutions. This makes the test more representative of real-world problems. In this way, ISPSO was tested to see how well the algorithm managed to recognize solutions without any information about them. For the purpose of comparisons with other methods, ISPSO was given the number of global and local minima, and all



(f) Two-dimensional $F6$



(g) Two-dimensional $F7$

Figure 8: Test functions for scalability test.

the runs were terminated immediately when the algorithm claimed that it had found exactly the same number of solutions as given. In real applications, the algorithm may have to be terminated at a maximum number of iterations because even the number of solutions is not known a priori.

3.3.1. Experimental setup for performance test

ISPSO has four control parameters for particle movement ($|S|$, r_{species}, r_{prey}, and r_{nest}) and three nesting parameters (a, ϵ_f, and ϵ_x). The default swarm size was set to $|S| = 20$, the default value used by Brits et al. (2007). The default species radius was set to $r_{\text{species}} = 0.1L$, as suggested by Li (2004) for one-dimensional problems, where L is the diagonal length of the search space, which is defined as

$$L = \sqrt{\sum_{i=1}^{D} (x_{\max,i} - x_{\min,i})^2} \tag{3.22}$$

where $x_{\max,i}$ and $x_{\min,i}$ are the i^{th} dimensional upper and lower bounds of the search space, respectively. The default nest radius was set to $r_{\text{nest}} = 0.01L$, 1% of the default species radius, and the default prey radius was set to $r_{\text{prey}} = 0.0001L$, 1% of the default nest radius. Particle movement was controlled with the cognitive and social coefficients, and the constriction factor as defined in Eqs. (3.3) and (3.4). $\psi_1 = \psi_2 = 2.05$ was used as recommended by Eberhart and Shi (2000) and Clerc and Kennedy (2002), and $\vec{v}_{\max}$ was set to a small value of $0.1 \cdot (\vec{x}_{\max} - \vec{x}_{\min})$ to test the exploration capability of quasi-random sampling. Note that, by limiting the velocity of particles, explorability is mainly controlled by preying and preemptive nesting. Newly created particles are assigned a non-zero random velocity constrained to be within the maximum initial velocity, $|\vec{v}_{\max,0}|$, to prevent early stagnation. In this test, the maximum initial velocity of $|\vec{v}_{\max,0}| = 0.001L$ was used.

Four sets of experiments were performed, each of which tested the sensitivity of the performance of ISPSO to each control parameter. One parameter was changed at a time with the others fixed to the default values. Each test function was optimized 30 times with the given value of each parameter, and the statistics of these results were reported.

The age parameter was set to $a = 10$ for all runs. In order to compare the results of ISPSO with those of NichePSO and SPSO reported by Brits et al. (2007) and Li (2004), respectively, ϵ_f was fixed to 10^{-4}, and ϵ_x was set to 10^{-3} according to Liong and Muttil (2001). However, note that Brits et al. (2007) and Li (2004) used a threshold value of 10^{-4} to stop test runs by measuring the closeness of a particle to the corresponding true minimum while ISPSO used ϵ_f and ϵ_x to measure how stable a species seed is. The closeness of a solution found to the true minimum was separately measured after optimization, as described below.

The criteria used to judge whether or not ISPSO has successfully found global

and local optima must be carefully defined. The algorithm cannot discard previous non-best solutions as it can for unimodal problems. In the current test, any solutions falling outside of the pre-specified radius of the true solutions were regarded as failures. The radii from the true solutions used in this study were 1%, 2%, 3%, 4%, and 5% of the diagonal spanning of the search space defined in Eq. (3.22) (hereafter referred to as the accuracy). Since one solution set is evaluated against the five accuracies after optimization, if all the solutions satisfy a certain accuracy, those solutions also satisfy the lower accuracies.

3.3.2. Experimental setup for scalability test

The scalability test was performed with $F6$ and $F7$ which have an exponentially growing number of solutions with increasing dimensions. For each dimension of the functions, 10 individual optimization runs were carried out, and their results were summarized. The values of the control parameters were set to $|S|$ = the number of solutions, $r_{\text{species}} = 0.1L$, $r_{\text{prey}} = 0.001L$, and $r_{\text{nest}} = 0.05L$ while the same nesting criteria as in the performance test was used as follows: $a = 10$, $\epsilon_f = 10^{-4}$, and $\epsilon_x = 10^{-3}$. Each run was terminated immediately when the algorithm claimed to find all the solutions or reached the maximum number of iterations, whichever happened first. The maximum number of iterations was set to 8,000 so that the maximum allowed number of function evaluations was 8,000 times the number of solutions, the same criteria used in Brits et al. (2007) except for the case of the 1-dimensional $F6$ function, where 6,000 more function evaluations were allowed in this study to be consistent with the other cases. The maximum number of function evaluations required to find all the solutions of the 1-dimensional $F6$ function was checked to see whether or not ISPSO actually evaluated the function more than NichePSO did in Brits et al. (2007).

3.4. Results and discussion

3.4.1. Performance

The performance and the consistency of ISPSO, NichePSO, and SPSO are compared in Table 10. The performance was measured with the number of function evaluations required for convergence and the consistency with the success rate of finding all solutions. The NichePSO and SPSO algorithms were not run again in this study. Instead, the values reported by Brits et al. (2007) and Li (2004) were used for comparison. A swarm size of 20 was used with ISPSO and NichePSO, and 50 with SPSO. ISPSO was evaluated at 1% accuracy. Note that SPSO did not try to find local optima, so the results of SPSO for $F2$ and $F4$ were not comparable with those of the other algorithms. The computational cost of ISPSO was consistently lower than that of SPSO for $F1$, $F3$, and $F5$, and ISPSO was comparable with SPSO in terms of the standard deviation of the computational cost. ISPSO succeeded to consistently find all the solutions with fewer function evaluations on average than NichePSO except for $F5$. For $F1$, $F3$, and $F5$, where all the solutions have nearly the same fitness value, ISPSO showed more variability than NichePSO in the number of function evaluations. For the other test functions, ISPSO outperformed NichePSO in terms of both performance and consistency. Assuming that the number of function evaluations follows the normal distribution, a t-test was carried out to statistically compare the results of ISPSO and NichePSO on $F5$. The one-tailed p-value of the t-test was 0.273, which suggests that even though the average number of function evaluations required by ISPSO is lower than for NichePSO, the performances of ISPSO and NichePSO are not significantly different at a 95% confidence level. Overall, ISPSO shows improved performance over SPSO and significant improvement over NichePSO. The sensitivity of ISPSO performance to each control parameter is discussed below.

Table 10: Comparison of ISPSO's performance with those of NichePSO and SPSO. [a] As reported by Brits et al. (2007). [b] As reported by Li (2004) (rounded off). * Not available because SPSO found only global optima.

Function	Algorithm	Number of function evaluations	Success rate (%)
$F1$	ISPSO	$1,074 \pm 285$	100
	NichePSO[a]	$2,372 \pm 109$	100
	SPSO[b]	$1,383 \pm 243$	100
$F2$	ISPSO	931 ± 218	100
	NichePSO[a]	$2,934 \pm 475$	93
	SPSO[b]	*	*
$F3$	ISPSO	$1,038 \pm 316$	100
	NichePSO[a]	$2,404 \pm 195$	100
	SPSO[b]	$1,248 \pm 319$	100
$F4$	ISPSO	909 ± 212	100
	NichePSO[a]	$2,820 \pm 517$	93
	SPSO[b]	*	*
$F5$	ISPSO	$2,203 \pm 426$	100
	NichePSO[a]	$2,151 \pm 200$	100
	SPSO[b]	$3,155 \pm 402$	100

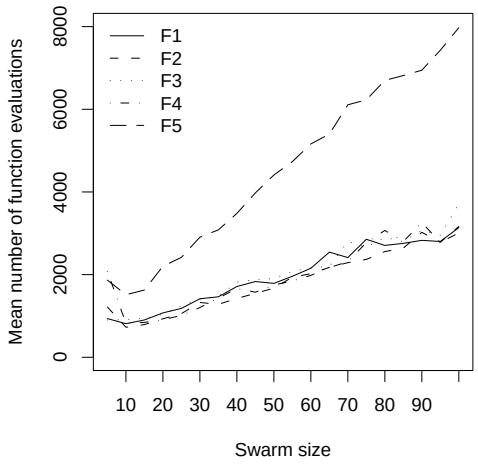
a. Swarm size

Swarm size $|S|$ was increased in increments of 5 from 5 to 100. ISPSO successfully located all the global and local minima within 1% accuracy except for $F5$ with a swarm size 10. For $F5$ with $|S| = 10$, the mean number of solutions within 1% accuracy was 3.93 (98.25%), and all the solutions were located within 3% accuracy (100%). The standard PSO 2007 C code (Clerc, 2007) suggests the following formula for swarm size:

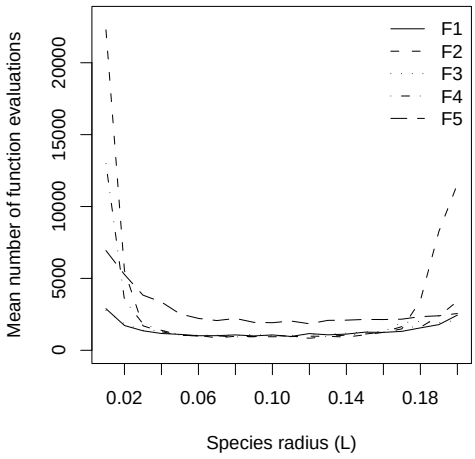
$$|S| = 10 + \left\lfloor 2\sqrt{D} \right\rfloor . \tag{3.23}$$

According to Eq. (3.23), a swarm size of 12 is recommended for global optimization of two-dimensional problems, such as $F5$. Using $|S| = 15$, ISPSO located all the solutions of $F5$ within 1% accuracy. This result shows that ISPSO does not require large swarm sizes for multi-modal problems and does not depend highly on the number of solutions. In comparison, NichePSO requires $|S| \geq N^2$ where N is the number of solutions (Brits et al., 2007), and SPSO requires $|S| = 50$ to obtain similar results (Li, 2004).

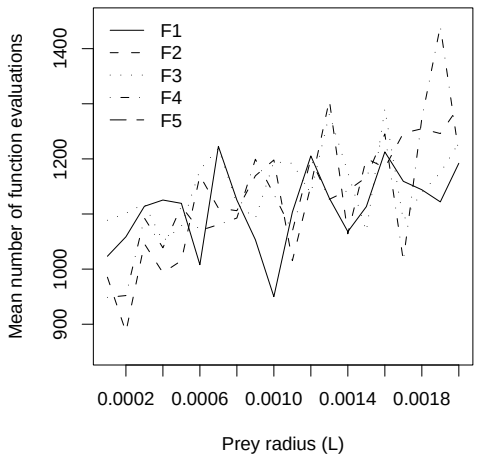
Figure 9a shows the mean number of function evaluations as a function of the swarm size. As can be seen in Figure 9a, swarm size 5 requires higher number of function evaluations than swarm size 10. At least two particles are required for non-isolated speciation, but, usually, more than two individuals have to be involved to actively explore the search space. For this reason, with a small number of particles, nesting takes place sequentially, and more iterations are needed to converge to multiple solutions. With a large swarm, multiple species can be generated at each iteration, which helps the swarm perform parallel searches. However, the function has to be evaluated for each individual in the swarm at every iteration before any nesting can take place. This behavior delays the evaluation of nesting criteria and,



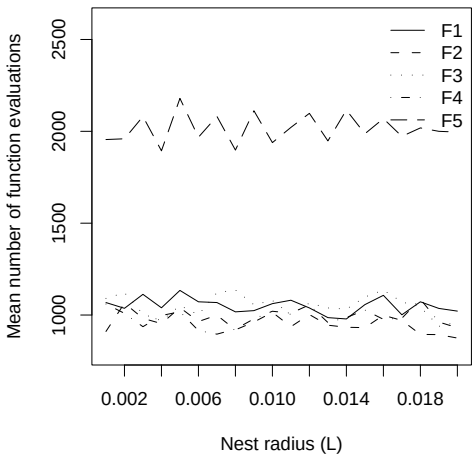
(a) Swarm size: 1% accuracy.



(b) Species radius: 2% accuracy.



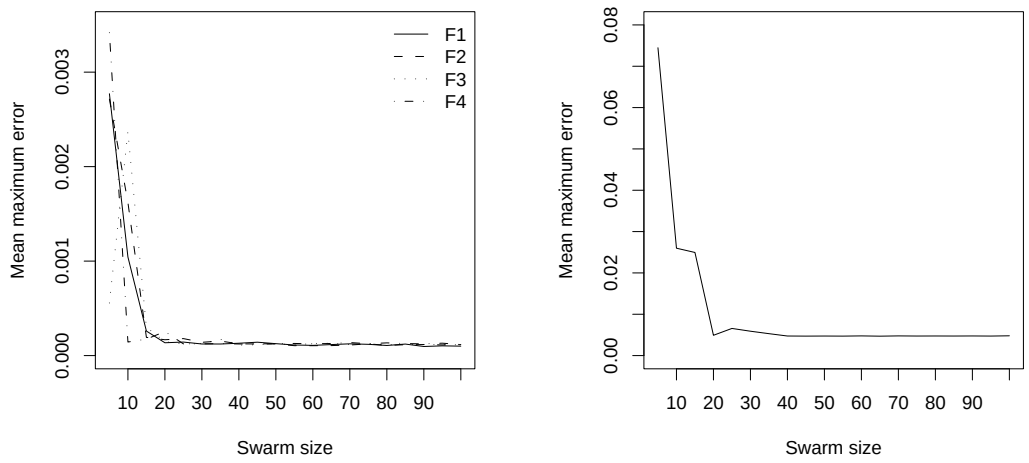
(c) Prey radius: 1% accuracy.



(d) Nest radius: 1% accuracy.

Figure 9: Computational cost versus different control parameters. Optimization runs with success rate of 100% at each accuracy are reported as mean values of 30 test runs.

hence, leads to higher computational costs. The trade-off between the computational costs of sequential and parallel searches is shown in Figure 9a. The swarm size of 10 corresponded to the lowest mean number of function evaluations for all functions. This $|S|$ value is nearly the same as calculated by Eq. (3.23) (i.e., $|S| = 12$ for $D = 1$ and $D = 2$), but it should be noted that Eq. (3.23) is used for global optimization, and multi-modal optimization may require more particles.



(a) $F1$–$F4$ (b) $F5$

Figure 10: Error versus swarm size. Optimization runs with success rate of 100% at 1% accuracy are reported as mean values of 30 test runs.

Figure 10 plots the mean value of maximum errors over 30 runs against the swarm size. Maximum error is the largest error value among those corresponding to multiple solutions. As indicated in Figure 10, maximum error stabilizes at a swarm size of 30 for $F1$ to $F4$ and 40 for $F5$. This result shows that using more particles yields more accurate answers. However, note that gain is not significant especially when computational cost is taken into account.

b. Species radius

Species radius was increased at $0.01L$ intervals from $0.01L$ to $0.20L$. Too small of species radius prevents the swarm from finding solutions because it decreases the chance of non-isolated speciation to take place. Species radius greater than $0.02L$ showed consistent performance. Although premature nesting was observed with $F5$ at 1% accuracy as shown in Figure 11, all solutions were found within 2% accuracy.

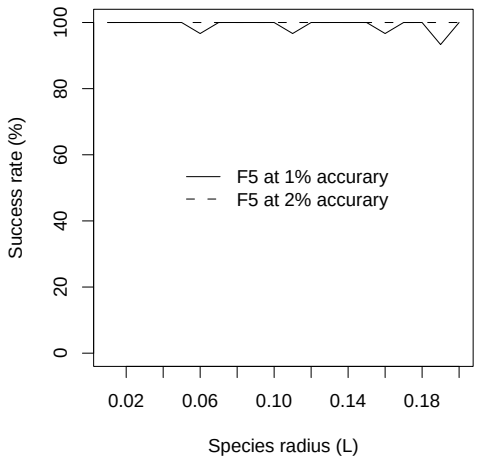


Figure 11: Success rate of finding all solutions versus species radius. Mean values of 30 test runs are reported.

Because speciation at each iteration takes place with a higher fitness value before with lower ones, the swarm has difficulty to generate species near local minima with relatively low fitness values when using small species radii. For functions having almost equal minima, all the solutions have nearly the same probability of generating species near them even with a small species radius, and this behavior significantly reduces computational cost. Note that, for $F2$ and $F4$, which have one global and four local minima, the success rates of $r_{\text{species}} = 0.01L$ are 80.00% and 76.67%, respectively, while the success rate of the same species radius for $F1$, $F3$, and $F5$ is 100%.

As shown in Figure 9b, species radii from $0.05L$ to $0.15L$ show low computational cost for $F1$ to $F4$, and the number of function evaluations for $F5$ starts stabilizing around $r_{\text{species}} = 0.10L$. For all of the test functions, species radii of $0.10L$ lead to high efficiencies. $r_{\text{species}} = 0.10L$ is approximately 50% of the shortest distance between neighboring solutions for all the functions. These results suggests that it is computationally efficient to use the species radius of 50% of the shortest distance between solutions. However, note that, in real-world applications, the species radius has to be pre-specified without knowing distances between solutions a priori, and needs to be carefully defined because only the fittest solution is obtained among any other local solutions within its species radius.

c. Prey radius

Since the prey radius is supposed to be a private region or volume of each particle, it remains small compared to species radius. Prey radius was increased by $0.0001L$ increments from $0.0001L$ to 0.002. As shown in Figure 9c, the greater the prey radius of the swarm, the greater the number of function evaluations needed to find the same number of solutions. The increasing uniformity of particle samples is achieved at the expense of slow convergence because particles clustering around a possible solution get scattered again. The smaller the prey radius, the faster the convergence while the greater the prey radius, the more the quasi-random sampling takes place. Therefore, the balance between fast convergence and sampling uniformity can be controlled by the prey radius, but it needs to be kept small to make convergence happen.

d. Nest radius

Nest radius varied from $0.001L$ to $0.020L$ at $0.001L$ intervals. As shown in Figure 9d, computational cost was not particularly sensitive to the nest radius. However, when

the maximum particle velocity was increased to $\vec{x}_{\max} - \vec{x}_{\min}$, the difference between the numbers of function evaluations for $r_{\text{nest}} = 0.001L$ and $r_{\text{nest}} = 0.020L$ clearly appeared. Because particles close to existing nests are replaced with new particles sampled from the Sobol' sequence, the bigger the nest radius, the more likely individuals in the swarm are to escape from the regions of existing nests, and the more uniformly the swarm explores the search space. This behavior increases the probability of finding local solutions and, hence, reduces computational cost. Table 11 presents the computational cost of two tests with different nest radii (i.e., $r_{\text{nest}} = 0.001L$ and $r_{\text{nest}} = 0.020L$). Each test consists of 30 separate optimization runs with $|S| = 20$ and $\vec{v}_{\max} = \vec{x}_{\max} - \vec{x}_{\min}$, and computational cost is in the format of $a \pm b$, where a and b are the mean and standard deviation of the number of function evaluations, respectively. Until finding three solutions, both nest radii showed similar performance, but, after then, individuals in the swarm with the smaller nest radius tend to keep flying around solutions already found, requiring considerably more function evaluations, because of the low probability of escaping from existing nests. In general, a swarm with a small nest radius requires more function evaluations to find the same number of solutions than one with a bigger nest radius does. However, the nest radius needs to be kept smaller than the species radius to ensure finding multiple solutions close to each other.

Table 11: Number of function evaluations for two test runs on $F5$ with different nest radii.

Solutions found	$r_{\text{nest}} = 0.001L$	$r_{\text{nest}} = 0.02L$
1	$1,163 \pm 287$	$1,283 \pm 378$
2	$1,477 \pm 282$	$1,680 \pm 323$
3	$1,968 \pm 405$	$2,038 \pm 481$
4	$4,017 \pm 2,164$	$2,926 \pm 881$

3.4.2. Scalability

Table 12 shows the results of scalability test on $F6$ and $F7$. The algorithm consistently found all the solutions for low-dimensional problems (i.e., the 1- and 2-dimensional $F6$, and the 1-dimensional $F7$). However, premature nesting was observed at 1% accuracy when the number of solutions exponentially increased along with the problem dimension. The percentages of the solutions found went up to $100 \pm 0\%$ at 4% accuracy except for the 5-dimensional $F6$. In 1 of the 10 experiments on $F6$ with $D = 5$, the algorithm failed to locate 2 solutions within the maximum number of function evaluations at 5% accuracy (i.e., 99.18%). The number of function evaluations required to find all the solutions was approximately $20 \cdot \mathrm{NS}^2$ for $F6$ and $7 \cdot \mathrm{NS}^2$ where NS is the number of solutions. When the number of function evaluations is divided by the maximum number of iterations (i.e., 8,000 in this study), the required swarm size $|S| = c \cdot \mathrm{NS}^d$ is obtained, where c is 2.5×10^{-3} for $F6$ and 8.75×10^{-4} for $F7$, and d is 2. This relationship between the swarm size and the number of solutions agrees well with Brits et al. (2007). However, ISPSO requires only a faction of the swarm size of NichePSO to obtain better results. For example, NichePSO used $|S| = 4 \cdot \mathrm{NS}$ while ISPSO used $|S| = \mathrm{NS}$ for the same test functions. Note that the maximum number of actual function evaluations in the case of the 1-dimensional $F6$ was 705 and, in (Brits et al., 2007), the maximum allowed number of function evaluations in the same case was 18,000; therefore, ISPSO did not consume more function evaluations than NichePSO even if 6,000 more function evaluations were allowed in the test.

As shown in Table 13, a comparison between ISPSO and NichePSO suggests that ISPSO consistently outperformed NichePSO in massively multi-modal problems (e.g., the number of solutions of $F6$ up to 243). Note that the results for $F7$ (i.e., the Griewank function) were not compared because the number of solutions analytically

Table 12: Results of the scalability test on $F6$ and $F7$. Percentage of the solutions found was evaluated at 1% accuracy, and the number of function evaluations of 10 experiments are reported regardless of their number of the solutions found.

Function	Dimension	Solutions	% solutions found	Number of function evaluations
$F6$	1	3	100 ± 0	475 ± 162
	2	9	100 ± 0	$2,534 \pm 388$
	3	27	99.26 ± 1.56	$16,160 \pm 1,800$
	4	81	98.77 ± 1.01	$112,809 \pm 17,575$
	5	243	98.07 ± 0.34	$1,210,407 \pm 392,340$
$F7$	1	5	100 ± 0	633 ± 134
	2	31	98.06 ± 2.26	$11,455 \pm 1,252$
	3	157	95.54 ± 1.67	$174,615 \pm 13,702$

and numerically found in this study does not match that presented in Brits et al. (2007).

Table 13: Comparison of the percentage of the solutions found of $F6$ between ISPSO and NichePSO. [a] As reported by Brits et al. (2007).

Dimension	ISPSO (1% accuracy)	NichePSO[a]
1	100	100
2	100	100
3	99.26	97.45
4	98.77	97.08
5	98.07	92.00

In the Griewank function, $F7(\vec{x})$ is greater than or equal to the parabolic function $\frac{1}{4000}\sum_{j=1}^{D} x_j^2$ because $\prod_{j=1}^{D} \cos\left(\frac{x_j}{\sqrt{j}}\right) \in [-1, 1]$. One global minimum is located at $\vec{0}$, and local minima exist near the tangent points where the Griewank function meets

the parabolic function. The following conditions hold true at global and local minima:

$$F7'_i(\vec{x}) = \frac{x_i}{2000} + \frac{\sin\left(\frac{x_i}{\sqrt{i}}\right)}{\sqrt{i}} \cdot \prod_{j=1,\, j\neq i}^{D} \cos\left(\frac{x_j}{\sqrt{j}}\right) = 0 \quad \text{for } i = 1, \cdots, D \tag{3.24}$$

$$F7''_i(\vec{x}) = \frac{1}{2000} + \frac{1}{i} \cdot \prod_{j=1}^{D} \cos\left(\frac{x_j}{\sqrt{j}}\right) > 0 \quad \text{for } i = 1, \cdots, D \tag{3.25}$$

where $F7'_i(\vec{x})$ and $F7''_i(\vec{x})$ are the first and the second partial derivatives of the Griewank function $F7(\vec{x})$ with respect to x_i, respectively. The tangential slope of a point (i.e., $|F7'_i(\vec{x})|$) tends to increase along the $\frac{x_i}{2000}$ line as the dimension i or $|x_i|$ increases. Because $\prod_{j=1,\, j\neq i}^{D} \cos\left(\frac{x_j}{\sqrt{j}}\right) \in [-1, 1]$, no points eventually satisfy Eq. (3.24) when $|x_i|$ is greater than a certain threshold value. Although it is trivial to visually count the number of minima of the Griewank function in up to 2-dimensional problems, it is not straightforward to analytically count the number of minima because it is difficult to find the threshold value for $|x_i|$ due to high correlation between dimensions in high-dimensional problems. This issue is addressed in detail in Chapter IV. In Brits et al. (2007), 5, 25, and 625 minima were found for the 1-, 2-, and 3-dimensional problems in the search space $[-28, 28]^D$, respectively, while 9, 111, and 1,215 were found for the same cases in this study. This discrepancy in the number of minima in the search space prevented a comparison of the percentage of the solutions found. However, a visual inspection of the 1-dimensional search space confirmed that the number of minima in $[-28, 28]$ is 9 as shown in Figure 12. I did not attempt to reproduce the results from the NichePSO implementation of Brits et al. (2007) due to uncertainty about the details of the implementation. I present the results for the Griewank function $F7$ from only ISPSO as another test of the scalability of the algorithm. Additional work would be required to compare these results directly with those from the NichePSO implementation of Brits et al. (2007).

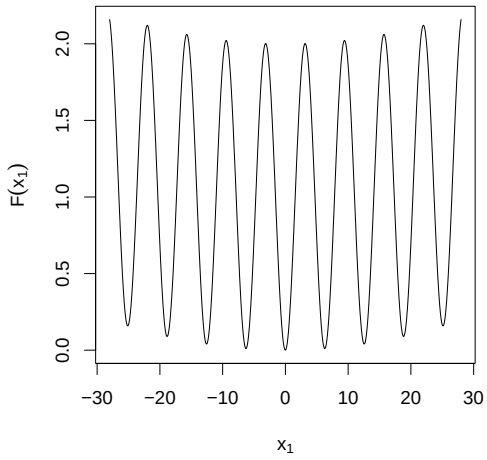


Figure 12: One-dimensional Griewank function in $[-28, 28]$. Note that the number of minima is 9.

3.5. Summary and conclusions

The Species-based PSO (SPSO) was modified for multi-modal problems by employing "isolated speciation." Particles left alone after proximity-based speciation form an additional species called "the isolated species" to make themselves move around the search space. This simple modification of SPSO guarantees more dynamic speciation of particles and reduces swarm size required to locate all solutions. This behavior helps reduce unnecessary function evaluations that do not contribute to finding solutions. In addition, the exploration of the search space was improved through deterministic low-discrepancy sampling during optimization. This sampling takes place through exclusion of particles from the small regions of known solutions and fitness assimilation between particles in contact. The exclusive regions of already found solutions also prevent unnecessary function evaluations near them and increase sampling uniformity. This approach is called the Isolated Speciation-based PSO (ISPSO).

Widely used multi-modal functions were used to test the performance and the

scalability of ISPSO. The results were compared with the reported works of SPSO (Li, 2004) and NichePSO (Brits et al., 2007), and the comparison has shown that, in most cases, ISPSO outperformed SPSO and NichePSO in terms of computational cost, consistency, and scalability. However, ISPSO experienced increasing difficulty with massively multi-model problems in locating all solutions at high accuracy; particles tended to converge prematurely, which is a common problem in PSOs (Lv et al., 2006). This premature convergence needs to be addressed in future work. More reliable criteria for the detection of solutions may help in this regard. Computational complexity required at each iteration needs to be improved. In Chapter V, a modification of the ISPSO algorithm was applied to a high-dimensional distributed hydrologic model to perform uncertainty analysis.

CHAPTER IV

DERIVATION OF THE NUMBER OF MINIMA OF THE GRIEWANK FUNCTION

4.1. Introduction

The Griewank function (Griewank, 1981) has been widely used to test the convergence of optimization algorithms (Kennedy, 2000; Krink et al., 2002; Riget and Vesterstrøm, 2002; Xie et al., 2002; Brits et al., 2003; Locatelli, 2003; Khemka and Jacob, 2004; He et al., 2004; Acan and Gunay, 2005; Monson and Seppi, 2005; Meissner et al., 2006; Monson and Seppi, 2006; Brits et al., 2007; Wang et al., 2007) because it has an exponentially growing number of minima with increasing dimensions (Locatelli, 2003; Brits et al., 2007). The function is defined as follows:

$$f_n(\vec{x}) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{4.1}$$

within $[-600, 600]^n$ where n is the dimension of the function. The search space is usually restricted to $[-x_{\max}, x_{\max}]^n$, where $x_{\max} > 0$, and the global minimum is located at $\vec{x} = \vec{0}$ with a value of 0. The actual number of minima may not be important when a global optimization is conducted. However, the number of minima needs to be well defined to test any technique that searches for local optima. Most studies vaguely mention the number of minima of the function (Locatelli, 2003; Khemka and Jacob, 2004; He et al., 2004) and, to the best of the author's knowledge, no analytical derivation of the number of minima of the Griewank function has been given in the literature. However, knowing the number of minima is critical if the Griewank function is to serve as the basis for evaluating algorithms designed to find local minima as well as global ones (i.e., multi-modal optimization). In some cases, the

number of solutions given is inconsistent with analytical results. For example, Brits et al. (2007) compared the ability of NichePSO, nbest PSO, lbest PSO, sequential niching, and deterministic crowding based on the number of minima found through numerical searches. However, further work with the ISPSO algorithm introduced in Chapter III has found a different number of solutions than found by Brits et al. (2007). In order to address this issue and provide a consistent basis for comparing algorithms, this chapter analytically derives the number of minima of the Griewank function. An approach is developed in three basic steps: (1) the search space is restricted to a hyperrectangle, (2) it is shown that the hyperrectangle is the maximum possible hyperrectangle within which the local minimum on the Griewank function correspond to the tangent points on a simpler surface, and (3) an analytical approach is developed for finding the number of tangent points on the simpler surface. This approach yields an accurate count of the number of local minima of the Griewank function within the defined hyperrectangle.

Section 4.2 elaborates on the characteristics of the function's surface and the high correlation between dimensions in problems of more than one dimension, and redefines the problem of counting the number of minima to make it analytically tractable. However, because of the complicated nature of the function's surface and the high degree of correlation between dimensions, the domain space needs to be restricted to hyperrectangles found by the numerical method introduced in Section 4.2. Although the analytical method derived in Section 4.3 cannot be applied to arbitrary domain spaces, it should be noted that the method does not miss any minima within hyperrectangles satisfying certain conditions. As most optimization algorithms are tested within fixed hyperrectangles, it remains practical to use hyperrectangles as domain spaces for testing many optimization algorithms.

4.2. Redefinition of the problem

If the partial derivative of a function has a simple form, the number of maxima and minima of the function could be derivable by setting the derivative functions to 0. However, the partial derivative of the Griewank function with respect to x_i has a fairly complex form as follows:

$$\frac{\partial f_n(\vec{x})}{\partial x_i} = \frac{x_i}{2000} + \frac{\sin\left(\frac{x_i}{\sqrt{i}}\right)}{\sqrt{i}} \cdot \prod_{j=1,\, j\neq i}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right) \tag{4.2}$$

and it is difficult, if not impossible, to analytically solve this non-linear system involving n variables. Global and local minima have to satisfy the following conditions:

$$f'_{n,i}(\vec{x}) = \frac{x_i}{2000} + \frac{\sin\left(\frac{x_i}{\sqrt{i}}\right)}{\sqrt{i}} \cdot \prod_{j=1,\, j\neq i}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right) = 0 \quad \text{for } i = 1, \cdots, n \tag{4.3}$$

$$f''_{n,i}(\vec{x}) = \frac{1}{2000} + \frac{1}{i} \cdot \prod_{j=1}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right) > 0 \quad \text{for } i = 1, \cdots, n \tag{4.4}$$

where $f'_{n,i}(\vec{x})$ and $f''_{n,i}(\vec{x})$ are the first and second derivatives of $f_n(\vec{x})$, respectively. Note that i is an index for dimensions, not an imaginary number. Eq. (4.4) is required to ensure that maxima are not taken into account. By rearranging Eq. (4.4), we obtain $\prod_{j=1}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right) > -\frac{i}{2000}$. Because the region of non-positive values of $\prod_{j=1}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right)$ satisfying Eqs. (4.3) and (4.4) (i.e., $f_n(\vec{x}) \geq \frac{1}{4000}\sum_{j=1}^{n} x_j^2 + 1$ at local minima) is outside of the region of its positive values (i.e., $f_n(\vec{x}) < \frac{1}{4000}\sum_{j=1}^{n} x_j^2 + 1$ at local minima), problem domains in this study are restricted such that

$$\prod_{j=1}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right) > 0. \tag{4.5}$$

Since a value of $\frac{i}{2000}$ is small for low dimensions, not much portion of the function space is lost. Eq. (4.3) can be rewritten as an equality of two functions as follows:

$$\sin\left(\frac{x_i}{\sqrt{i}}\right) = -\frac{x_i\sqrt{i}}{2000}\left[\prod_{j=1,\, j\neq i}^{n}\cos\left(\frac{x_j}{\sqrt{j}}\right)\right]^{-1} \tag{4.6}$$

where $\prod_{j=1,\, j\neq i}^{n}\cos\left(\frac{x_j}{\sqrt{j}}\right) \neq 0$ because $\prod_{j=1}^{n}\cos\left(\frac{x_j}{\sqrt{j}}\right) > 0$.

Because $f_n(\vec{x})$ meets the surface of $\frac{1}{4000}\sum_{j=1}^{n} x_j^2$ at the global minimum and near local minima, we will find the tangent points of $f_n(\vec{x})$ by finding the tangent points of the simpler surface $\frac{1}{4000}\sum_{j=1}^{n} x_j^2$ and deriving the relationship between these two sets of tangent points. Further references to tangent points in this chapter are defined as tangent points on the surface $\frac{1}{4000}\sum_{j=1}^{n} x_j^2$ unless otherwise noted. As shown in Figure 13, local minima and tangent points lie on different projected surfaces because the slope of the tangent plane (i.e., $f'_{n,i}(\vec{x})$) at a tangent point is not 0 except at $\vec{x} = \vec{0}$. Since we only want to know the number of minima, their exact coordinates are not of direct interest. In this chapter, the number of minima is indirectly derived by counting the number of tangent points associated with them. Because the tangent point associated with the global minimum is the global minimum itself, this method also takes into account the global minimum. Therefore, problem domains have to be carefully defined so that there always exists one minimum corresponding to one tangent point. As i or x_i increases, $f'_{n,i}(\vec{x})$ also tends to increase along the $\frac{x_i}{2000}$ line and, eventually, points satisfying Eq. (4.3) disappear, which makes global optimization easier (Locatelli, 2003). Because there are correlations between independent variables in problems of more than one dimension, it is hard to say whether or not there are local minima satisfying $0 < \prod_{j=1}^{n}\cos\left(\frac{x_j}{\sqrt{j}}\right) < 1$, by looking at $f'_{n,i}(\vec{x})$ surfaces separately. It is necessary to know the maximum extent of each x_i beyond which there are no local minima associated with tangent points. For $n = 1$, it is trivial to check the

maximum extent of x_1 because all the points lie on $f'_{1,1}(x_1)$. For $n \geq 2$, some type of numerical analysis is required to find the corners of a hyperrectangle (i.e., a domain space) where a local minimum with the flattest slope possible is located. Beyond the edges of the hyperrectangle defined by the corners, there could be tangent points not associated with local minima. In the following sections, the simplest case of $n = 1$ is explained, and a numerical method to find the maximum domain space for $n \geq 2$ is introduced.

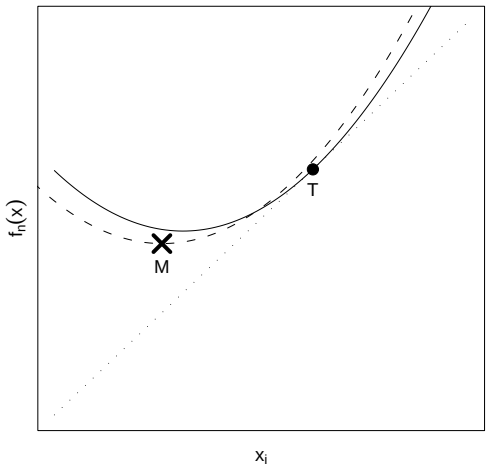


Figure 13: One-dimensional schematic showing difference between a local minimum M and its corresponding tangent point T. x_i is greater than 0. The solid and dashed lines represent the Griewank function $f_n(\vec{x})$ with different x_j $(1 \leq j \leq n,\ j \neq i)$ sets, and the dotted line represents $\frac{1}{4000}\sum_{j=1}^{n} x_j^2$.

4.2.1. One-dimensional case

Figure 14 shows where the last two local minima exist in the one-dimensional Griewank function space. Note that, while tangent points are evenly distributed at every 2π, local minima are not, and each tangent point has its own corresponding local minimum. If the boundary of a domain space is located between a tangent point and

its corresponding local minimum, the number of tangent points is not the same as the number of local minima. For this reason, a problem domain U is defined for the moment such that $U = (0,\, 2\pi k)$ where $k \in N$. The maximum value of k, $k_{\max}$, is defined such that the largest local minimum associated with a tangent point is located between $(2\pi(k_{\max} - 1),\, 2\pi k_{\max})$. Figure 14 clearly shows that $k_{\max}$ for $n = 1$ is $\left\lfloor \frac{600}{2\pi} \right\rfloor = 95$, and a domain space defined by $k > 95$ is extended to beyond the domain of the function definition (i.e., $2\pi \times 96 > 600$). Therefore, the number of local minima in the domain space $U = (0,\, 2\pi k)$ is k where $1 \leq k \leq k_{\max}$.

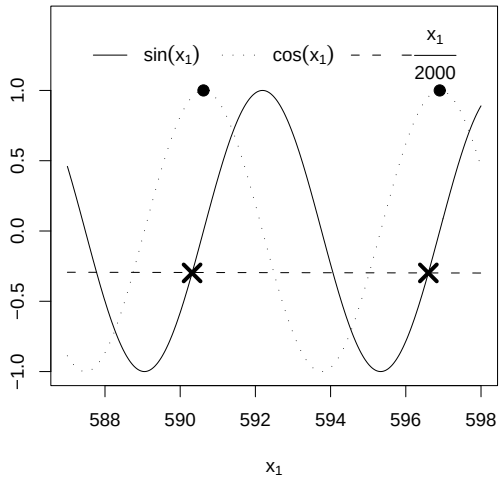


Figure 14: Last two local minima in a one-dimensional problem. The black dots are tangent points, and the "X" symbols are local minima.

For the sake of generality, a method of finding $k_{i,\max}$ (i.e., $k_{\max}$ for the i^{th} axis) in an arbitrary one-dimensional domain space (i.e., $n = 1$) defined by the i^{th} axis of high-dimensional problems as shown in Figure 15 is introduced. It is obvious that local minima exist between $(2\pi\sqrt{i}k_i - \frac{1}{2}\pi\sqrt{i},\, 2\pi\sqrt{i}k_i)$, where $k_i \in N$, because $\prod_{j=1,\, j\neq i}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right) = 1$ in Eq. (4.6) ($\because n = 1$), and Eq. (4.5) is satisfied only within this region. The first local minimum can be found in $(\frac{3}{2}\pi\sqrt{i},\, 2\pi\sqrt{i})$, if any. Using the periodicity of the sine curve, the second local minimum can be also found in

$(\frac{3}{2}\pi\sqrt{i},\, 2\pi\sqrt{i})$ by shifting the straight line defined by the right-hand side of Eq. (4.6). The $k_i{}^{\text{th}}$ local minimum, $\vec{x}^{k_i} = (x_1^{k_i}, \cdots, x_n^{k_i})$, is obtained by solving the following shifted version of Eq. (4.6):

$$\sin\left(\frac{{x'_i}^{k_i}}{\sqrt{i}}\right) = -\frac{{x'_i}^{k_i}\sqrt{i} + 2\pi i(k_i - 1)}{2000}\left[\prod_{j=1,\, j\neq i}^{n}\cos\left(\frac{x_j^{k_i}}{\sqrt{j}}\right)\right]^{-1} \tag{4.7}$$

where ${x'_i}^1 = x_i^1$ and ${x'_i}^{k_i} = x_i^{k_i} - 2\pi\sqrt{i}(k_i - 1)$. For one-dimensional problems, Eq. (4.7) is further simplified by setting $n = 1$ and $\prod_{j=1,\, j\neq i}^{n}\cos\left(\frac{x_j}{\sqrt{j}}\right) = 1$. There are no local minima at $x'_i = \frac{3}{2}\pi\sqrt{i}$ if the straight line in the right-hand side of Eq. (4.7) meets the sine curve at $\left(\frac{3}{2}\pi\sqrt{i}, \sin\left(\frac{3}{2}\pi\sqrt{i}\right)\right)$ as follows:

$$-\frac{\frac{3}{2}\pi i + 2\pi i(\alpha - 1)}{2000}\left[\prod_{j=1,\, j\neq i}^{n}\cos\left(\frac{x_j}{\sqrt{j}}\right)\right]^{-1} = -1 \tag{4.8}$$

where $\alpha \in R$ such that $k_{i,\max} = \lfloor\alpha\rfloor$, and $\lfloor\cdot\rfloor$ is the maximum integer less than or equal to a given number (i.e., the flooring function). By solving Eq. (4.8), we get $k_{i,\max} = \lfloor\alpha\rfloor = \left\lfloor \frac{1000}{\pi i}\cdot\prod_{j=1,\, j\neq i}^{n}\cos\left(\frac{x_j}{\sqrt{j}}\right) + \frac{1}{4}\right\rfloor$. However, since the Griewank function is defined within $[-600, 600]^n$, $2\pi\sqrt{i}k_{i,\max}$ must be less than or equal to $x_{\max} = 600$. In summary, $k_{i,\max}$ is obtained by

$$k_{i,\max} = \min\left(\left\lfloor\frac{x_{\max}}{2\pi\sqrt{i}}\right\rfloor,\, \left\lfloor\frac{1000}{\pi i}\cdot\prod_{j=1,\, j\neq i}^{n}\cos\left(\frac{x_j}{\sqrt{j}}\right) + \frac{1}{4}\right\rfloor\right) \tag{4.9}$$

and, given a one-dimensional domain space $(0, 2\pi\sqrt{i}k_i)$ where $1 \leq k_i \leq k_{i,\max}$, k_i is the number of local minima.

4.2.2. Multi-dimensional case

In problems of more than one dimension, because the position of a local minimum in one axis is highly correlated with those in the other axes, it is not trivial to analytically solve Eq. (4.7) for all dimensions. The values of $\cos\left(\frac{x_i}{\sqrt{i}}\right)$ and $k_{i,\max}$ for $i = 1, \cdots, n$

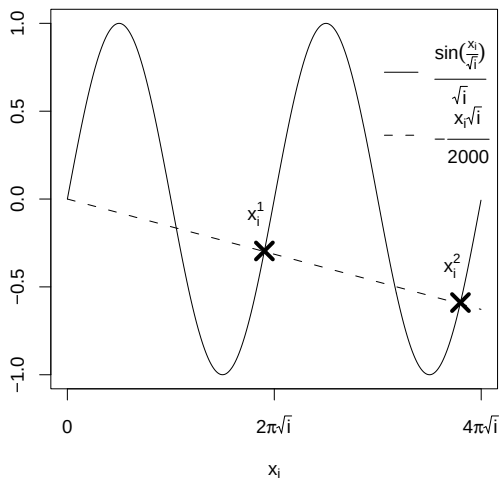


Figure 15: Two local minima in a one-dimensional problem defined by the i^{th} axis of high-dimensional problems. Not scaled for explanations.

can be numerically estimated as described in Figure 16. The subroutine defined in Figure 17 is used to solve Eq. (4.7) for each dimension at a time. Because the range of $x_i'^k$ is $(\frac{3}{2}\pi\sqrt{i},\, 2\pi\sqrt{i})$ and, given an estimated value of $\prod_{j=1,\, j\neq i}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right)$, there is only one solution to Eq. (4.7), if any, within this range, any optimization algorithm can estimate $x_i'^k$. However, $x_i'^k$ found in this way may not be the correct one because the correlation between dimensions is not taken into account when solving Eq. (4.7). The estimated value of $x_i'^k$ is used to evaluate $\prod_{j=1,\, j\neq i}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right)$, which is iteratively plugged into Eq. (4.7) to estimate the next value of $x_i'^k$.

Once $k_{i,\max}$ is estimated, a problem domain needs to be well defined. Define a problem domain by $U = (0,\, x_{i,\max})$, where $0 < x_{i,\max} \leq 2\pi\sqrt{i}k_{i,\max}$, such that $x_{i,\max}$ does not have to be $2\pi\sqrt{i}k_i$ where $1 \leq k_i \leq k_{i,\max}$. When $\prod_{j=1,\, j\neq i}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right)$ is greater than 0, a local minimum is found in the following range:

$$x_i \in \left(2\pi\sqrt{i}k_i - \frac{1}{2}\pi\sqrt{i},\, 2\pi\sqrt{i}k_i\right) \tag{4.10}$$

Require: $n \geq 1$ {Problem dimension}
Require: ϵ_f {Training threshold for $f'_{n,i}(\vec{x}_{k_{i,\max}})$}
Require: ϵ_c {Training threshold for $\cos\left(\frac{x_i}{\sqrt{i}}\right)$}
Require: $\text{iter}_{\max}$ {Maximum number of iterations for $\cos\left(\frac{x_i}{\sqrt{i}}\right)$}
  $x_{\max} \Leftarrow 600$ {Initial domain}
  $\vec{c}_{\text{out}} \Leftarrow \vec{1}$ {n-tuple output vector}
  repeat
    for $i = 1, \cdots, n$ **do**
      $\vec{c}_{\text{tr}} \Leftarrow \vec{1}$ {n-tuple training vector}
      for $\text{iter} = 1, \cdots, \text{iter}_{\max}$ **do**
        $x_i^{k_{i,\max}} \Leftarrow$ **getxi**$(\vec{c}_{\text{tr}}, i)$ in Figure 17
        $c_{\text{tr},i} \Leftarrow \cos\left(\frac{x_i^{k_{i,\max}}}{\sqrt{i}}\right)$
        if $\text{iter} > 1$ and $|c_{\text{tr},i} - c_{\text{prev}}| < \epsilon_c$ **then**
          break
        $c_{\text{prev}} \Leftarrow c_{\text{tr},i}$
        for $j = 1, \cdots, n,\ j \neq i$ **do**
          $c_{\text{tr},j} \Leftarrow \cos\left(\frac{\textbf{getxi}(\vec{c}_{\text{tr}},j)}{\sqrt{j}}\right)$
      $c_{\text{out},i} \Leftarrow c_{\text{tr},i}$
    if $\left|f'_{n,i}(\vec{x}_{k_{i,\max}})\right| \leq \epsilon_f\ \forall i = 1, \cdots, n$ **then**
      break
    $x_{\max} \Leftarrow x_{\max} - 2\pi$
  until $x_{\max} \leq 0$
  return $\vec{c}_{\text{out}}$

Figure 16: Pseudo code to estimate $\cos\left(\frac{x_i}{\sqrt{i}}\right)$ for $i = 1, \cdots, n$.

Require: $n \geq 1$ {Problem dimension}
Require: $x_{\max} > 0$ {Problem domain}
Require: $\vec{c}_{\text{in}}$ {Input: $\cos\left(\frac{x_i}{\sqrt{i}}\right)$ values}
Require: i {Input: the current training dimension}
Require: All other variables are local ones.
  for $i' = 1, \cdots, n$ **do**
    $\cos\left(\frac{x_{i'}}{\sqrt{i'}}\right) \Leftarrow c_{\text{in},i'}$
  calculate $k_{i,\max}$ according to Eq. (4.9)
  estimate $x_i'^{k_{i,\max}}$ by solving Eq. (4.7)
  $x_i^{k_{i,\max}} \Leftarrow x_i'^{k_{i,\max}} + 2\pi\sqrt{i}(k_{i,\max} - 1)$
  return $x_i^{k_{i,\max}}$

Figure 17: Pseudo code for the **getxi** subroutine.

because $\cos\left(\frac{x_i}{\sqrt{i}}\right)$ is greater than 0, satisfying Eq. (4.5), and Eq. (4.6) can hold true only in this range. Likewise, when $\prod_{j=1,\, j\neq i}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right)$ is less than 0, a local minimum is found in the following range:

$$x_i \in \left(2\pi\sqrt{i}k_i - \frac{3}{2}\pi\sqrt{i},\, 2\pi\sqrt{i}k_i - \pi\sqrt{i}\right). \tag{4.11}$$

$x_{i,\max}$ needs to avoid these ranges because, otherwise, it is possible that local minima found in the above ranges cannot be associated with the tangent points at $x_i = 2\pi\sqrt{i}k_i \pm \pi\sqrt{i}$, which means that the analytical method introduced in this chapter cannot be applied. Therefore, the allowable ranges of $x_{i,\max}$ are as follows:

$$\left[2\pi\sqrt{i}k_i - 2\pi\sqrt{i},\, 2\pi\sqrt{i}k_i - \frac{3}{2}\pi\sqrt{i}\right] \tag{4.12}$$

or

$$\left[2\pi\sqrt{i}k_i - \pi\sqrt{i},\, 2\pi\sqrt{i}k_i - \frac{1}{2}\pi\sqrt{i}\right]. \tag{4.13}$$

The initial condition for $x_{i,\max}$, Eq. (4.12), and Eq. (4.13) can be interpreted as

$$0 < x_{i,\max} \leq 2\pi\sqrt{i}k_{i,\max} \tag{4.14}$$

and

$$x_{i,\max} \in X = \left\{ x_i \;\middle|\; x_i \text{ is a multiple of } \frac{\pi}{2}\sqrt{i} \;\vee\; \left\lfloor \frac{x_i}{\frac{\pi}{2}\sqrt{i}} \right\rfloor \text{ is an even integer} \right\}. \quad (4.15)$$

In case where $x_{i,\max}$ does not satisfy Eq. (4.15) because integer values for $x_{i,\max}$ are usually preferred, we need to make sure that there are no local minima in

$$\left(\left\lfloor \frac{x_{i,\max}}{\frac{\pi}{2}\sqrt{i}} \right\rfloor \cdot \frac{\pi}{2}\sqrt{i},\; x_{i,\max} \right) \quad (4.16)$$

where $0 < x_{i,\max} \leq 2\pi\sqrt{i}k_{i,\max}$ and $x_{i,\max} \notin X$. This test can be done indirectly by checking whether or not the distance in the i^{th} axis between $x_{i,\max}$ and the closest tangent point whose coordinate is greater than $x_{i,\max}$ is greater than the possibly largest distance between them. Define the closest tangent point whose coordinate is greater than x_i as

$$t_i(x_i) = \left\lceil \frac{x_i}{\frac{\pi}{2}\sqrt{i}} \right\rceil \cdot \frac{\pi}{2}\sqrt{i}. \quad (4.17)$$

The largest distance between local minima and their corresponding tangent points can be obtained by calculating $t_i(x_i^{k_{i,\max}}) - x_i^{k_{i,\max}}$ because $t_i(x_i^k)$ is the tangent point associated with x_i^k, and the distance between them also increases as x_i increases. If $t_i(x_{i,\max}) - x_{i,\max}$ is greater than $t_i(x_i^{k_{i,\max}}) - x_i^{k_{i,\max}}$, there must be one local minimum in $(x_{i,\max}, t_i(x_{i,\max}))$ along the i^{th} axis, which means that there are no local minima in the range defined by Eq. (4.16).

When $x_{i,\max}$ satisfies all the requirements described above, a domain space can be extended to $U = [-x_{i,\max},\, x_{i,\max}]\ \forall i \in [1, n]$ because the negative domain space of $(-x_{i,\max},\, 0)$ is symmetrical to $(0,\, x_{i,\max})$, and the analytical method derived in the following section takes into account both regions implicitly.

4.3. Derivation of the number of minima

The following theorems are used throughout this section:

Theorem 4.3.1 *If $x \in R$, $\lfloor x \rfloor = \left\lfloor \frac{x}{2} \right\rfloor + \left\lfloor \frac{x}{2} + \frac{1}{2} \right\rfloor$ where $\lfloor x \rfloor$ is the maximum integer less than or equal to x (i.e., the flooring function of x).*

Proof Let $x = i + r$ such that $i = \lfloor x \rfloor$ and $0 \leq r < 1$. If $i = 2j$ where $j \in N$, the following equalities hold:

$$\left\lfloor \frac{x}{2} \right\rfloor = \left\lfloor \frac{2j + r}{2} \right\rfloor = \left\lfloor j + \frac{r}{2} \right\rfloor = j \tag{4.18}$$

$$\left\lfloor \frac{x}{2} + \frac{1}{2} \right\rfloor = \left\lfloor \frac{2j + r}{2} + \frac{1}{2} \right\rfloor = \left\lfloor j + \frac{r}{2} + \frac{1}{2} \right\rfloor = j \tag{4.19}$$

because $0 \leq \frac{r}{2} < \frac{1}{2}$ and $\frac{1}{2} \leq \frac{r}{2} + \frac{1}{2} < 1$. From Eqs. (4.18) and (4.19), the following holds for even $\lfloor x \rfloor$'s:

$$\lfloor x \rfloor = \left\lfloor \frac{x}{2} \right\rfloor + \left\lfloor \frac{x}{2} + \frac{1}{2} \right\rfloor = 2j = i. \tag{4.20}$$

If $i = 2j + 1$ where $j \in N$, the following equalities hold:

$$\left\lfloor \frac{x}{2} \right\rfloor = \left\lfloor \frac{2j + 1 + r}{2} \right\rfloor = \left\lfloor j + \frac{r}{2} + \frac{1}{2} \right\rfloor = j \tag{4.21}$$

$$\left\lfloor \frac{x}{2} + \frac{1}{2} \right\rfloor = \left\lfloor \frac{2j + 1 + r}{2} + \frac{1}{2} \right\rfloor = \left\lfloor j + \frac{r}{2} + 1 \right\rfloor = j + 1 \tag{4.22}$$

because $\frac{1}{2} \leq \frac{r}{2} + \frac{1}{2} < 1$ and $0 \leq \frac{r}{2} < \frac{1}{2}$. From Eqs. (4.21) and (4.22), the following holds for odd $\lfloor x \rfloor$'s:

$$\lfloor x \rfloor = \left\lfloor \frac{x}{2} \right\rfloor + \left\lfloor \frac{x}{2} + \frac{1}{2} \right\rfloor = 2j + 1 = i. \tag{4.23}$$

Therefore, from Eqs. (4.20) and (4.23), $\lfloor x \rfloor = \left\lfloor \frac{x}{2} \right\rfloor + \left\lfloor \frac{x}{2} + \frac{1}{2} \right\rfloor$, as required. ∎

Theorem 4.3.2 *If $x \in R$, $\lceil x \rceil = \left\lceil \frac{x}{2} \right\rceil + \left\lceil \frac{x}{2} - \frac{1}{2} \right\rceil$ where $\lceil x \rceil$ is the minimum integer greater than or equal to x (i.e., the ceiling function of x).*

Proof Let $x = i - r$ such that $i = \lceil x \rceil$ and $0 \leq r < 1$. If $i = 2j$ where $j \in N$, the following equalities hold:

$$\left\lceil \frac{x}{2} \right\rceil = \left\lceil \frac{2j - r}{2} \right\rceil = \left\lceil j - \frac{r}{2} \right\rceil = j \tag{4.24}$$

$$\left\lceil \frac{x}{2} - \frac{1}{2} \right\rceil = \left\lceil \frac{2j - r}{2} - \frac{1}{2} \right\rceil = \left\lceil j - \frac{r}{2} - \frac{1}{2} \right\rceil = j \tag{4.25}$$

because $0 \leq \frac{r}{2} < \frac{1}{2}$ and $\frac{1}{2} \leq \frac{r}{2} + \frac{1}{2} < 1$. From Eqs. (4.24) and (4.25), the following holds for even $\lceil x \rceil$'s:

$$\lceil x \rceil = \left\lceil \frac{x}{2} \right\rceil + \left\lceil \frac{x}{2} - \frac{1}{2} \right\rceil = 2j = i. \tag{4.26}$$

If $i = 2j + 1$ where $j \in N$, the following equalities hold:

$$\left\lceil \frac{x}{2} \right\rceil = \left\lceil \frac{2j + 1 - r}{2} \right\rceil = \left\lceil j - \frac{r}{2} + \frac{1}{2} \right\rceil = j + 1 \tag{4.27}$$

$$\left\lceil \frac{x}{2} - \frac{1}{2} \right\rceil = \left\lceil \frac{2j + 1 - r}{2} - \frac{1}{2} \right\rceil = \left\lceil j - \frac{r}{2} \right\rceil = j \tag{4.28}$$

because $-\frac{1}{2} \leq \frac{r}{2} - \frac{1}{2} < 0$ and $0 \leq \frac{r}{2} < \frac{1}{2}$. From Eqs. (4.27) and (4.28), the following holds for odd $\lceil x \rceil$'s:

$$\lceil x \rceil = \left\lceil \frac{x}{2} \right\rceil + \left\lceil \frac{x}{2} - \frac{1}{2} \right\rceil = 2j + 1 = i. \tag{4.29}$$

Therefore, from Eqs. (4.26) and (4.29), $\lceil x \rceil = \left\lceil \frac{x}{2} \right\rceil + \left\lceil \frac{x}{2} - \frac{1}{2} \right\rceil$, as required. ∎

In the previous section, the problem was redefined so that counting the number of tangent points is the same as counting the number of minima. The cosine function is defined between $[-1, 1]$ and, thus, the range of the function (i.e., part of Eq. (4.1))

$$\prod_{j=1}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right) \tag{4.30}$$

is also restricted to $[-1, 1]$. Consequently, $1 - \prod_{j=1}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right)$ has values between $[0, 2]$ and Eq. (4.1) between $\left[\frac{1}{4000}\sum_{j=1}^{n} x_j^2,\ \frac{1}{4000}\sum_{j=1}^{n} x_j^2 + 2\right]$. Therefore, any tangent

points of Eq. (4.1) lies on the surface of $\frac{1}{4000}\sum_{j=1}^{n} x_j^2$ when Eq. (4.30) is 1.

The absolute value of $\cos\left(\frac{x_i}{\sqrt{i}}\right)$ is 1 when x_i is a multiple of $\pi\sqrt{i}$. How many 1's of $\left|\cos\left(\frac{x_i}{\sqrt{i}}\right)\right|$ exist depends on the range of x_i or $[x_{i,\min}, x_{i,\max}]$. The number of $\pi\sqrt{i}k$'s within this range, where $k \in N$, is calculated as

$$N_i = \left\lfloor \frac{x_{i,\max}}{\pi\sqrt{i}} \right\rfloor - \left\lceil \frac{x_{i,\min}}{\pi\sqrt{i}} \right\rceil + 1. \tag{4.31}$$

The number of x_i's satisfying $\cos\left(\frac{x_i}{\sqrt{i}}\right) = 1$ is calculated as

$$N_i^+ = \left\lfloor \frac{x_{i,\max}}{2\pi\sqrt{i}} \right\rfloor - \left\lceil \frac{x_{i,\min}}{2\pi\sqrt{i}} \right\rceil + 1 \tag{4.32}$$

and the number of x_i's satisfying $\cos\left(\frac{x_i}{\sqrt{i}}\right) = -1$ is derived as

$$N_i^- = N_i - N_i^+ = \left\lfloor \frac{x_{i,\max}}{2\pi\sqrt{i}} + \frac{1}{2} \right\rfloor - \left\lceil \frac{x_{i,\min}}{2\pi\sqrt{i}} - \frac{1}{2} \right\rceil \tag{4.33}$$

according to Theorems 4.3.1 and 4.3.2. Now, the number of maxima and minima can be expressed as

$$M_n = \prod_{j=1}^{n} N_j. \tag{4.34}$$

Counting the number of the n-tuples in the set

$$A_n = \left\{ \left(\cos\left(\frac{x_1}{\sqrt{1}}\right), \cdots, \cos\left(\frac{x_n}{\sqrt{n}}\right) \right) \in [-1,1]^n \;\middle|\; \prod_{j=1}^{n} \cos\left(\frac{x_j}{\sqrt{j}}\right) = 1 \right\} \tag{4.35}$$

is a combinatorial problem where combinations take place without repetitions. It is obvious that any element, $\cos\left(\frac{x_i}{\sqrt{i}}\right)$, of an n-tuple belonging to the set A_n must have a value of -1 or 1 because, otherwise, the absolute value of Eq. (4.30) cannot be 1. Because Eq. (4.30) should be 1, an even number of elements in the n-tuple have a value of -1, and the other elements have a value of 1. Therefore, the number of the

n-tuples in the set A_n can be expressed as

$$\sum_{j=0}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{2j} = \sum_{j=0}^{\lfloor \frac{n}{2} \rfloor} \frac{n!}{(n-2j)!(2j)!} \tag{4.36}$$

where $\binom{n}{2j}$ is the binomial coefficient. Encode an n-tuple in A_n as

$$(a_1, a_2, \cdots, a_n) \tag{4.37}$$

where a_i is either 1 or -1. If 1 and -1 are substituted with the + and − symbols, respectively, n-tuples in A_n can be represented as (+, +, $\cdots$, +), (−, −, +, $\cdots$, +), (−, +, −, +, $\cdots$, +) (i.e., one n-tuple of $\binom{n}{0}$ and two examples of $\binom{n}{2}$, respectively), and so on. Note that there are an even number of the − symbols, and the others are all +'s. The numbers of x_i values satisfying $a_i = +$ and $a_i = -$ are N_i^+ and N_i^-, respectively.

Counting all the possible vector $\vec{x}$'s generating the n-tuples in the set A_n can be done recursively in terms of n. The simplest form is $S_1 = N_1^+$ for $n = 1$ where S_n is the number of minima for dimension n. For $n = 2$, there are S_1 minima when a_2 is fixed to + because S_1 number of x_1's satisfying $\prod_{j=1}^{n-1} a_j = +$ also satisfy $\left(\prod_{j=1}^{n-1} a_j\right) a_n = \prod_{j=1}^{n} a_j = +$. If a_2 is fixed to −, $\prod_{j=1}^{n-1} a_j$ must be −, and the number of a_1 satisfying this condition is $M_1 - S_1$ (i.e., the number of maxima for $n = 1$). Therefore, $S_2 = S_1 \cdot N_2^+ + (M_1 - S_1) \cdot N_2^-$. Generalizing this recursive form, the following equations are obtained:

$$S_1 = N_1^+ \quad \text{if n=1,} \tag{4.38}$$

$$S_n = S_{n-1} \cdot N_n^+ + (M_{n-1} - S_{n-1}) \cdot N_n^- \quad \text{if } n > 1 \tag{4.39}$$

for $[-x_{i,\min},\, x_{i,\max}]\ \forall i \in [1,n]$. Now, Eqs. (4.38) and (4.39) can be expanded as follows:

$$S_1 = \left\lfloor \frac{x_{1,\max}}{2\pi} \right\rfloor - \left\lceil \frac{x_{1,\min}}{2\pi} \right\rceil + 1 \quad \text{if n=1,} \tag{4.40}$$

$$\begin{aligned} S_n = {} & S_{n-1} \cdot \left(\left\lfloor \frac{x_{n,\max}}{2\pi\sqrt{n}} \right\rfloor - \left\lceil \frac{x_{n,\min}}{2\pi\sqrt{n}} \right\rceil + 1 \right) \\ & + \left[\prod_{j=1}^{n-1} \left(\left\lfloor \frac{x_{j,\max}}{\pi\sqrt{j}} \right\rfloor - \left\lceil \frac{x_{j,\min}}{\pi\sqrt{j}} \right\rceil + 1 \right) - S_{n-1} \right] \\ & \times \left(\left\lfloor \frac{x_{n,\max}}{2\pi\sqrt{n}} + \frac{1}{2} \right\rfloor - \left\lceil \frac{x_{n,\min}}{2\pi\sqrt{n}} - \frac{1}{2} \right\rceil \right) \quad \text{if } n > 1 \end{aligned} \tag{4.41}$$

for $[-x_{i,\min},\, x_{i,\max}]\ \forall i \in [1,n]$.

4.4. Results and discussion

Figure 18 and Table 14 present the maximum estimated number of local minima, $k_{i,\max}$, and the largest local minimum $x_i^{k_{i,\max}}$ on the i^{th} axis. They define hyperrectangles within which Eqs. (4.40) and (4.41) can be applied. Outside these regions, the analytical method introduced in this chapter cannot be used to count the number of minima. Figure 18 shows $k_{i,\max}$ for different dimensions. From $n = 43$, the numerical algorithm in Figure 16 experienced difficulties in finding $k_{i,\max}$, and no plots were drawn. This result might be caused by reducing the search space by 2π in all dimensions. However, as shown in Table 15, because the number of minima within only a small fraction of the hyperrectangles defined by $x_i^{k_{i,\max}}$ is so high even for $n = 4$ (e.g., 77,647 minima in $[-45, 45]^4$, a subspace of $[-x_i^{k_{i,\max}},\, x_i^{k_{i,\max}}]\ \forall i \in [1,4]$ for $n = 4$), it would be practically enough to define domain spaces for up to $n = 40$. Table 14 shows $k_{i,\max}$ and $x_i^{k_{i,\max}}$ estimated for up to four-dimensional problems. Note that $k_{i,\max}$ for the same i varies with n because of correlation between dimensions. When defining a domain space by $U = [-x_{i,\max},\, x_{i,\max}]\ \forall i \in [1,n]$, we need

to make sure $0 < x_{i,\max} \leq t_i(x_i^{k_i,\max})$. This condition satisfies Eq. (4.14) because $t_i(x_i^{k_i,\max}) = 2\pi\sqrt{i}k_{i,\max}$ for all the cases in Table 14. Also, $x_{i,\max}$ has to satisfy Eq. (4.15) or Eq. (4.16).

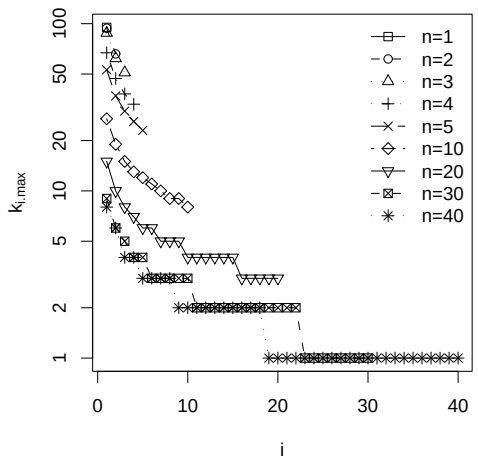


Figure 18: $k_{i,\max}$ versus i for different problem dimensions.

As a set of examples, domain spaces $U = [-x_{\max}, x_{\max}]^n$ were evaluated for $1 \leq n \leq 4$ where $x_{\max} \in \{14, 28, 45\}$. Note that, for the sake of simplicity, the domain spaces were chosen such that $x_{i,\max} = x_{\max}\ \forall i \in [1, n]$. For $x_{\max} = 14$, Eq. (4.15) holds true when $i = 1, 2$, or 4. The closest tangent point whose coordinate is greater than $x_{3,\max} = 14$ is $t_3(14)$, and the distance along the 3$^{\text{rd}}$ axis between $x_{3,\max}$ and $t_3(14)$ is $t_3(14) - 14 = 2.32$. This distance is greater than $t_3(x_3^{k_{3,\max}}) - x_3^{k_{3,\max}} = 1.24$ for $n = 3$ as shown in Table 14. This means that the local minimum associated with $t_3(x_{3,\max})$ exists in $(x_{3,\max}, t_3(x_{3,\max}))$, not in the range defined by Eq. (4.16) for $x_{3,\max} = 14$. For $x_{\max} = 28$, Eq. (4.15) holds true when $i = 2, 3$, or 4. A visual inspection of the x_1 axis and a numerical analysis show that there are no local minima in the range defined by Eq. (4.16) for $x_{1,\max} = 28$. For $x_{\max} = 45$, Eq. (4.15) holds true when $1 \leq i \leq 4$. Because $x_{\max} \in \{14, 28, 45\}$ satisfies the boundary conditions specified

Table 14: Maximum estimated number of local minima and largest local minimum in each dimension. $k_{i,\max}$ is the maximum estimated number of local minima on the i^{th} axis within $x_i \in (0, 600)$, $x_i^{k_{i,\max}}$ is the largest local minimum on the i^{th} axis, $t_i(x_i^{k_{i,\max}})$ is its corresponding tangent point, and $t_i(x_i^{k_{i,\max}}) - x_i^{k_{i,\max}}$ is the largest distance on the i^{th} axis between them.

n	i	$k_{i,\max}$	$x_i^{k_{i,\max}}$	$t_i(x_i^{k_{i,\max}})$	$t_i(x_i^{k_{i,\max}}) - x_i^{k_{i,\max}}$
1	1	95	596.60	596.90	0.30
2	1	94	590.28	590.62	0.34
	2	66	585.82	586.46	0.64
3	1	88	552.45	552.92	0.47
	2	62	550.04	550.92	0.88
	3	51	553.78	555.02	1.24
4	1	67	420.63	420.97	0.34
	2	47	416.98	417.63	0.65
	3	38	412.61	413.55	0.94
	4	33	413.47	414.69	1.22

by Eqs. (4.15) and (4.16), we can safely use Eqs. (4.40) and (4.41) to calculate the number of minima of the Griewank function. Table 15 shows the numbers of minima for the three different search spaces for up to four dimensions.

Table 15: Numbers of minima for $[-14, 14]^n$, $[-28, 28]^n$, and $[-45, 45]^n$.

n	$[-14, 14]^n$	$[-28, 28]^n$	$[-45, 45]^n$
1	5	9	15
2	31	111	305
3	157	1,215	5,177
4	787	10,989	77,647

4.5. Summary and conclusions

It is difficult to analytically solve the derivative of the Griewank function and directly count the number of minima because of the complicated nature of the function's sur-

face and the high degree of correlation between dimensions. The problem of counting the number of minima was redefined as counting the number of tangent points lying on the parabolic plane, which is part of the function. A numerical method was developed to find hyperrectangles that this approach can be applied to, and the number of minima of the function was analytically derived within these domain spaces based on the recursive functional form proposed in this chapter. The maximum extents of hyperrectangles for up to four dimensions were estimated, and the numbers of minima for three different search spaces were provided as a reference.

The numerical and analytical methods introduced in this chapter can be used to determine the exact number of minima within the domain space defined by a hyperrectangle satisfying certain conditions. The number of minima derived in this chapter can serve as a sound basis for evaluating multi-modal optimization algorithms.

CHAPTER V

APPLICATION OF MULTI-MODAL OPTIMIZATION TO GENERALIZED LIKELIHOOD UNCERTAINTY ESTIMATION

5.1. Introduction

Draper and Box (1987) stated "Remember that all models are wrong; the practical question is how wrong do they have to be to not be useful." For models to be useful, they have to be optimized in some way through parameter estimation, which typically requires calibration and validation (Thiemann et al., 2001). Uncertainty in the parameter estimates should be reduced by calibration to meaningfully estimate unobserved variables under hypothetical conditions (Koh et al., 2004). During the past two decades, many efforts have been devoted to developing automatic calibration methods to find optimal solutions (Thiemann et al., 2001). Due to the complexity of the objective function surface for distributed hydrologic models, classical gradient-descent-based algorithms are often trapped in local optima (Duan et al., 1992). The inability of the classical search algorithms to find the global optimum has led to the need for global optimization algorithms (Duan et al., 1992). Population-based algorithms were developed to avoid trapping into local optima and reliably locate the best parameter set. Among others, these include restarting gradient methods at random locations, simulated annealing (Kirkpatrick et al., 1983), genetic algorithms (Holland, 1962), and the Shuffled Complex Evolution (SCE-UA) algorithm (Duan et al., 1993). However, since there is no single measure to assess different aspects of the disagreement between the observed and simulated values, multiple objective functions have been employed to take advantages of various characteristics of different measures for the model performance (Gupta et al., 1998; Yapo et al., 1998). When one objective

function cannot be improved without compromising the others, the parameter set is called Pareto optimal (Yapo et al., 1998). However, there is no guarantee that models performing well in calibration will perform equally well in validation and prediction (Hassan, 2004). A single best parameter set or even multiple Pareto-optimal parameter sets are not necessarily good approximations of the real system because they tend to be greatly influenced by calibration data and assumptions made in calibration (Stephenson and Freeze, 1974; Beven, 2006a). Since fixed parameter sets do not account for the time-variant nature of reality, there remains much uncertainty in estimated parameters even after calibration (Thiemann et al., 2001; Devonec and Barros, 2002; Hong et al., 2005).

Recently, much attention has been paid to uncertainty estimation techniques. Among others, the Generalized Likelihood Uncertainty Estimation (GLUE) framework (Beven and Binley, 1992) has been widely used in environmental modeling (e.g., Beven and Binley, 1992; Freer et al., 1996; Aronica et al., 1998; Beven et al., 2000; Beven and Freer, 2001; Makowski et al., 2002; Muleta and Nicklow, 2005; Zheng and Keller, 2007) because of the few assumptions required and its ease of implementation (Schulz et al., 1999; Jacquin and Shamseldin, 2007). In the GLUE framework, a subjectively chosen likelihood measure is used to assess the model performance and weight the model output (Beven and Binley, 1992). The likelihood measure is also used to estimate the posterior likelihood measure distributions of the model parameters by updating their prior likelihood measure distributions (Beven and Binley, 1992; Beven and Freer, 2001). The likelihood measure is not limited to formal likelihood functions used in classical Bayesian inference, but rather is the degree of subjective belief about how well the model reproduces the observed data (Jacquin and Shamseldin, 2007). The subjectivity of GLUE has been often criticized in the literature (Gupta et al., 2003; Montanari, 2005; Mantovan and Todini, 2006), and formal Bayes-

ian inference techniques, such as Bayesian Recursive Estimation (BaRE) (Thiemann et al., 2001) and Bayesian Total Error Analysis (BaTEA) (Kavetski et al., 2003), have been introduced to estimate statistically meaningful prediction limits. However, these prediction limits may be inappropriate for non-ideal cases such as the followings: (1) no correct model structure exists, (2) the statistical characteristics of the model errors are not known a priori, and (3) various sources of uncertainty cannot be incorporated into a single formal likelihood function. In fact, Beven et al. (2007) show that wrong statistical assumptions for the error structure can cause biases in the parameter estimates and the prediction limits. For further discussion on the GLUE framework, refer to Thiemann et al. (2001), Beven and Young (2003), Gupta et al. (2003), Beven (2006b), Mantovan and Todini (2006), Andréassian et al. (2007), Beven et al. (2007), Hall et al. (2007), and Todini and Mantovan (2007).

One of the main disadvantages of the GLUE framework is that sufficient parameter samples are required to properly characterize the likelihood measure surface (Beven, 2006a). However, it is difficult to know in advance how many samples are needed to find enough parameter sets that well represent the real system (Beven, 2006a). Complex models generally require a long run time, and their high dimensionality also hinders the efficient application of the GLUE framework to uncertainty analysis. Because sampling strategies employed for the GLUE framework usually take parameter samples randomly from the search space (Beven and Binley, 1992), it is likely that many samples are taken from the regions of low likelihood measures. In addition, it is not guaranteed even after extensive sampling that there exists an identifiable unique solution in the samples, which means that one cannot expect to find a unique optimal solution like in traditional model optimization. The reason for the lack of a unique solution is a different approach to model calibration. The GLUE framework is based on the "equifinality" thesis, which does not accept the

concept of a unique solution, while traditional calibration methods pursue the best parameter set usually without accounting for the uncertainty in the predicted values (Beven and Binley, 1992; Beven and Freer, 2001; Beven, 2006a). Beven (2006a) suggests that there may be more than one acceptable model because of errors in the model structure and the measurement. Therefore, by design, the GLUE framework does not attempt to search for optimal solutions, and they are unlikely to be found in random samples. Sampling techniques, such as Markov chain Monte Carlo methods and importance sampling, are used to take samples from desired probability distributions in probabilistic Bayesian approaches. However, because the GLUE framework is not based on the assumption of statistical models for the error structure, these sampling techniques cannot be well incorporated into the framework. For this reason, the GLUE framework uses random sampling techniques and is computationally expensive to perform (Beven, 2006a).

The fundamental idea behind this study is that, by combining a multi-modal optimization algorithm and a uniform sampling technique, it would be possible within the GLUE framework, without statistical assumptions for the error structure, to find optimal solutions as well as to avoid sampling unnecessary parameter sets that perform poorly in terms of a subjective likelihood measure. The purpose of this study is not to evaluate the GLUE framework, but to incorporate a non-random sampling technique into the GLUE framework to reduce the computational burden required for the uncertainty analysis of costly distributed hydrologic models. This study investigates the feasibility of the application of a multi-modal optimization algorithm called the Isolated Speciation-based Particle Swarm Optimization (ISPSO) (see Chapter III for more details) to the uncertainty analysis of the Soil and Water Assessment Tool (SWAT) (Arnold et al., 1998) model output within the GLUE framework. To the best of the author's knowledge, no studies have been done to improve the computational ef-

ficiency of the GLUE framework by employing a multi-modal optimization algorithm. The proposed approach is expected to save much time required for uncertainty analysis because a run of the SWAT model is computationally very expensive. In this study, the prediction limits obtained by the proposed approach were compared with those of the GLUE approach with random sampling, and their sensitivities to different sample sizes were assessed. Section 5.2 briefly introduces the GLUE framework and the ISPSO algorithm, and describes how parameter samples taken in optimization are used to estimate prediction limits. Section 5.3 describes the likelihood measure used in this study and the SWAT model used to estimate streamflows and sediment discharges; provides data descriptions for a case study; and discusses results. Section 5.4 summarizes this chapter and draws conclusions.

5.2. Methodology

5.2.1. Generalized Likelihood Uncertainty Estimation

In the Generalized Likelihood Uncertainty Estimation (GLUE) framework (Beven and Binley, 1992), the term "model" includes the model parameters as well as the model structures (Beven, 2006a), and models are classified as either "behavioral" or "non-behavioral." When models are able to reproduce observed values to an acceptable level, they are classified as behavioral; otherwise, they are classified as non-behavioral. The criterion used to determine whether a model is behavioral or non-behavioral is its likelihood measure. If the model's likelihood measure is greater than a threshold value, the model is behavioral; otherwise, the model is non-behavioral.

Given a model structure, the GLUE framework updates the prior likelihood

measure of a set of the model parameters using the following equation:

$$L_{\text{posterior}}(\theta) = \frac{L(\theta|\xi, y)L_{\text{prior}}(\theta)}{C} \tag{5.1}$$

where $L_{\text{prior}}(\theta)$ and $L_{\text{posterior}}(\theta)$ denote the prior and posterior likelihood measures of the model parameter set θ, respectively, $L(\theta|\xi, y)$ is the likelihood measure of the model parameter set θ given the model input ξ and the observed data y corresponding to the model output, and C is a normalizing constant such that the sum of $L_{\text{posterior}}(\theta)$'s of all the parameter samples equals 1. Note that any goodness-of-fit measure can be used as the likelihood measure in the GLUE framework when its value varies from 0 to 1, and the perfect model is assigned a value of 1. That is, the likelihood measure is the modeler's degree of subjective belief about how much the model output is consistent with the observed data based on the objective function. Because the likelihood measure and the threshold value for behavioral models are subjectively chosen by the modeler, GLUE requires that those definitions should be made explicit so that those subjective choices and the results obtained using them can be examined and discussed more explicitly. The likelihood measure and the threshold value for behavioral models used in this study are defined later in Subsection 5.3.2.

The prediction quantiles of the behavioral models that simulate a value less than or equal to z_t at time t are evaluated as follows (Beven and Freer, 2001):

$$P(\hat{Z}_t < z_t) = \sum_{i=1}^{n} \left\{ L_{\text{posterior}}(\theta_i) \;\middle|\; \hat{Z}_{i,t} < z_t \right\} \tag{5.2}$$

where $\hat{Z}_{i,t}$ is the simulated value of Z at time t by the i^{th} behavioral model θ_i, and n is the number of the behavioral models. The prediction limits of the behavioral models at time t are calculated as follows: (1) weight the simulated value at time t of each behavioral model using its posterior likelihood measure, (2) sort the simulated

values of the behavioral models in ascending order, (3) accumulate the weights of the sorted simulated values in ascending order, and (4) define the lower and upper limits of the simulated values using the prediction quantiles. The prediction limits estimated in this way are referred to as "the GLUE uncertainty bounds" (Jacquin and Shamseldin, 2007; Zheng and Keller, 2007).

The GLUE uncertainty bounds are estimated from two percentiles of the weighted model outputs simulated by the behavioral models and do not have any probabilistic basis when a goodness-of-fit measure is used as the likelihood measure (Beven, 2006a; Montanari, 2007). The GLUE uncertainty bounds require the assumption that "the behavioural models in calibration will also be behavioural in prediction" (Beven, 2006a) and are also subjective because normalized subjective likelihood measures are used to weight the model outputs (McIntyre, 2004). This subjectivity in the GLUE uncertainty bounds is the main criticism often raised in the literature (Gupta et al., 2003; Montanari, 2005). Montanari (2005) discussed the subjective choices allowed in GLUE by examining several different likelihood measures and concluded that the GLUE uncertainty bounds are highly dependent on subjective choices such as the likelihood measure and the threshold value for behavioral models. The GLUE uncertainty bounds reflect the modeler's belief about what would be the ranges of the unobserved variables that the behavioral models are able to simulate with acceptable likelihood measures assessed in calibration. That is, when a behavioral model simulates observed data in calibration to a certain level of acceptability, the model is believed to simulate unobserved variables in prediction to the same level of acceptability. In this way, the unobserved variables simulated by the behavioral models are weighted according to their relative levels of acceptability (i.e., posterior likelihood measures), and certain percentiles of the simulated variables are determined as the GLUE uncertainty bounds. This rather crude interpretation of the GLUE uncer-

tainty bounds is due to the equifinality thesis arguing that there may be more than one acceptable model because of our imperfect knowledge about the system.

When we have the correct model structure, know the boundary conditions and the statistical characteristics of the model errors caused by uncertainty in the parameter estimates, and ignore measurement errors and parameter sampling errors, the prediction limits obtained by assuming a statistical error model (i.e., the probabilistic prediction limits) may provide statistically correct interpretations (Beven, 2006b). However, because models inevitably have structural errors in the equifinality thesis, and they actually do, it is questionable whether the probabilistic prediction limits will enclose unobserved variables with a given frequency when uncertainty in the model structure and the measurement is not taken into account explicitly.

That being said, the GLUE uncertainty bounds do not try to enclose a certain proportion of unobserved variables, but try to estimate certain percentiles of the weighted simulations that the behavioral models in the past would produce under hypothetical conditions (Beven, 2006a). This interpretation does not link the GLUE uncertainty bounds to a certain frequency with which unobserved variables will be found within the uncertainty bounds and gives the impression that the GLUE uncertainty bounds have nothing to do with future predictions. Rather than that, in the author's viewpoint, the GLUE uncertainty bounds represent the possible bounds of the simulated variables in prediction satisfying a certain level of subjective acceptability. Therefore, the GLUE uncertainty bounds highly depend on the likelihood measure and the threshold value for the behavioral models, which both reflect the modeler's perception about how well the model reproduces the data and how appropriately the likelihood measure evaluates the model performance. This is why the likelihood measure and the threshold value for the behavioral models should be made explicit for further discussion about the results.

5.2.2. Isolated Speciation-based Particle Swarm Optimization

A population-based heuristic algorithm called the Isolated Speciation-based Particle Swarm Optimization (ISPSO) (see Chapter III for more details) searches for global and local optima (i.e., multi-modal optimization), and tries to fill gaps between existing samples as uniformly as possible when it is allowed. In ISPSO, parameter samples at each iteration are referred to as particles, which are collectively called the swarm and move around in the search space toward preferable solutions. Deterministic uniform sampling increases the chance of discovering the regions of preferable solutions. A brief description of how ISPSO works is as follows: (1) particles form groups called species based on their fitness values and spatial proximity (i.e., speciation), (2) they are allowed to move in the search space with a pre-specified maximum velocity, (3) as iterations progress, their ages also increase, (4) once they are isolated from speciation, their ages are reset to 1, (5) old particles are assigned a high degree of trust, (6) old particles that converge to a certain point are considered solutions, and (7) steps 1 to 6 are repeated until stopping criteria are met. In this study, ISPSO was modified to adaptively update the maximum velocity according to particles' ages; this modification helps find optimal solutions in high-dimensional problems by stabilizing old particles' movements. The ISPSO algorithm uses a low-discrepancy sequence called the Sobol' sequence (Sobol', 1967) with the Owen (Owen, 1998) and Faure-Tezuka (Faure and Tezuka, 2002) scrambling schemes to uniformly generate samples, and has been implemented in the R language (R Development Core Team, 2006); a collection of R packages called Rmetrics (Würtz, 2004) was used to generate the scrambled Sobol' sequence.

5.2.3. Application of ISPSO to GLUE

In a real-number continuous problem space, a finite number of unevenly distributed parameter samples taken in global optimization may not well characterize the search space because most of the samples cluster around the global optimum. As a result, global optimization could miss many acceptable samples near local optima in high-dimensional search spaces. Compared to global optimization, multi-modal optimization searches for multiple local optima, and samples are likely to be taken from the vicinity of the local optima which are possibly equally acceptable in terms of the model performance. This feature of multi-modal optimization is better suited for the equifinality thesis than global optimization because the equifinality thesis also pursues multiple behavioral models. However, simply taking many samples around local optima could underestimate the value of less likely parameter sets in uncertainty analysis because those samples are not the direct interest of multi-modal optimization. ISPSO is a multi-modal optimization algorithm which finds local optima yet takes samples uniformly under certain conditions. Uniform samples contribute to the evaluation of inferior regions of the model parameters and help avoid the underestimation of less likely parameter sets.

In this study, the observed data are divided into three periods: a calibration period, a rejection period, and a verification period. The calibration period is used to take parameter samples from the search space by running one calibration run with ISPSO. The parameter samples that have a likelihood measure greater than a threshold value are referred to as behavioral models after calibration. The rejection period is used to reject part of the behavioral models after calibration when their likelihood measures are not greater than the threshold value. These rejected models fail to show consistent performance over the two different simulation periods. To test

the robustness of models in terms of model performance, the calibration and rejection periods should have quite different characteristics of the observed variables. If the two periods are similar, rejection may rarely reject models because most of them will perform equally well in both periods. The behavioral models after calibration that survived rejection are referred to as behavioral models after rejection. The behavioral models after rejection are used to estimate their GLUE uncertainty bounds of the variables in the verification period. Because the parameter samples taken by ISPSO are not randomly distributed, the approach introduced in this study is referred to as "the ISPSO-GLUE approach" as contrast to "the GLUE approach," whereby random samples are taken, and the GLUE uncertainty bounds obtained by this approach are referred to as "the ISPSO-GLUE uncertainty bounds." Note that the only difference between the two approaches is sampling techniques.

Because the optimal parameter sets in the objective function (i.e., goodness-of-fit measure) surface should have an optimal likelihood measure, the objective function used in optimization or the likelihood measure in the GLUE framework needs to be carefully defined to be correctly mapped to each other. For example, in minimization problems, a parameter sample with the lowest objective function value must have the highest likelihood measure. If the two surfaces are not properly mapped to each other, a calibration run with ISPSO cannot find parameter samples with a high likelihood measure. It should be also noted that the objective function is one-side bounded (e.g., $[0, \infty)$ in minimization problems) while the likelihood measure is two-side bounded (i.e., $[0, 1]$). Because the goodness-of-fit of the worst model cannot be well defined, optimization algorithms cannot use two-side bounded objective functions in calibration. Therefore, the likelihood measure surface should be a subspace of a transformed objective function surface (e.g., a truncated goodness-of-fit surface) because the infinite space of the objective function value cannot be directly mapped

to the finite space of the likelihood measure. For this reason, the objective function cannot be just the reciprocal of the likelihood measure.

Because the particles in ISPSO explore the search space comprehensively, the parameter samples generated in calibration cluster around global and local optima yet fill regions of low likelihood measures. These parameter samples are used to estimate the cumulative likelihood measure distributions of the model parameters after calibration. Some of them are rejected in rejection, and the models that survived rejection are used to estimate the cumulative likelihood measure distributions of the model parameters after rejection. The behavioral models after rejection are used to weight their simulated values and predict the ISPSO-GLUE uncertainty bounds in the verification period.

The behavioral models that perform consistently well in both calibration and rejection are evaluated with the verification data. At each time step, the simulated variables from all the behavioral models are ranked, and their corresponding likelihood measures are accumulated to estimate the cumulative likelihood measure distributions of the simulated variables. The lower and upper 2.5% percentiles of the cumulative likelihood measure distribution are discarded to build the 95% ISPSO-GLUE uncertainty bounds. Because the ISPSO-GLUE uncertainty bounds are assessed at each time step, no single parameter set is used to estimate either the upper or lower bound over the whole verification period. Note that, because the GLUE framework allows subjective likelihood measures instead of probabilistic likelihood functions, the ISPSO-GLUE uncertainty bounds are not probabilistic prediction limits, but the prediction limits of the behavioral models conditioned on the subjective likelihood measure and the threshold value used to identify the behavioral models. Because the GLUE uncertainty bounds are estimated from the cumulative likelihood measure distribution of the model output, it is not desirable to take too many samples

with a low likelihood measure (Jacquin and Shamseldin, 2007). If the frequency of model outputs with a low likelihood measure is so high that the cumulative likelihood measure distribution is significantly affected by these samples, it is unlikely to highlight enough model outputs with a high likelihood measure (Jacquin and Shamseldin, 2007). Because most of parameter sets are sampled by ISPSO in the vicinity of global and local optima, the high frequency of samples with a high likelihood measure will prevent giving too much weight to less likely model outputs. As a side effect, this approach will give much weight to highly likely model outputs compared to the GLUE approach, and the ISPSO-GLUE uncertainty bounds are expected to be narrower than the GLUE uncertainty bounds. In this study, a similar analysis was performed with random sampling of the same sample size to compare the results for the ISPSO-GLUE approach with those for the GLUE approach.

5.2.4. Evaluation of optimal models

Since ISPSO is able to find mathematically optimal model parameter sets (i.e., local optima or optimal models in the GLUE terminology) while exploring the search space, these models can be used for prediction without considering uncertainty in the context of traditional optimization. Optimal models are evaluated to see if fixed sets of the model parameters can be used to simulate different periods with different observed data. The purpose of this evaluation is to see if the model parameters are time-invariant as might be assumed by the model structure and if there exists a single set of the model parameters that can describes the watershed of interest.

5.3. Application

5.3.1. Study area and data descriptions

The Big Sandy Creek watershed shown in Figure 19 was used in this study. The area of the watershed is 598 km^2. A U.S. Geological Survey (USGS) gauge station is located at the outlet of the watershed. Daily streamflow and sediment discharge data are available from USGS (2006c) and USGS (2007), respectively, and were collected from October 1, 1984 to September 30, 1986. Weather data, including precipitation depths and temperatures, were obtained from NOAA-NCDC (2006) from October 1, 1983 to September 30, 1986 (i.e., one more year in addition to the same period for the daily streamflow and sediment discharge data). The National Elevation Dataset (NED) USGS (2006a) was used for watershed delineation, and the National Land Cover Dataset (NLCD) USGS (2006b) and the State Soil Geographic Database (STATSGO) (USDA-NRCS, 2006) were used to create HRUs. The watershed delineation was performed such that the generated stream network approximately matched the U.S. Environmental Protection Agency's Reach File 1 (RF1) data (USEPA, 2007). The land use distribution is urban 1.0%, agriculture 2.9%, forest 54.7%, rangeland 34.4%, and water/wetland 7.0%; the soil type distribution of the first layer of the STATSGO data is clay 9.1%, silt 19.6%, and sand 71.3%.

The two years of streamflow, sediment discharge, and weather data from October 1, 1984 to September 30, 1986 were divided into three periods: a calibration period from October 1, 1984 to May 31, 1985; a rejection period from June 1, 1985 to January 31, 1986; and a verification period from February 1, 1986 to September 30, 1986. Note that each simulation period is eight-month long due to the lack of daily sediment discharge data for the study area. Because SWAT requires an initial stabilization period to ensure that the output is not affected by the assumed initial

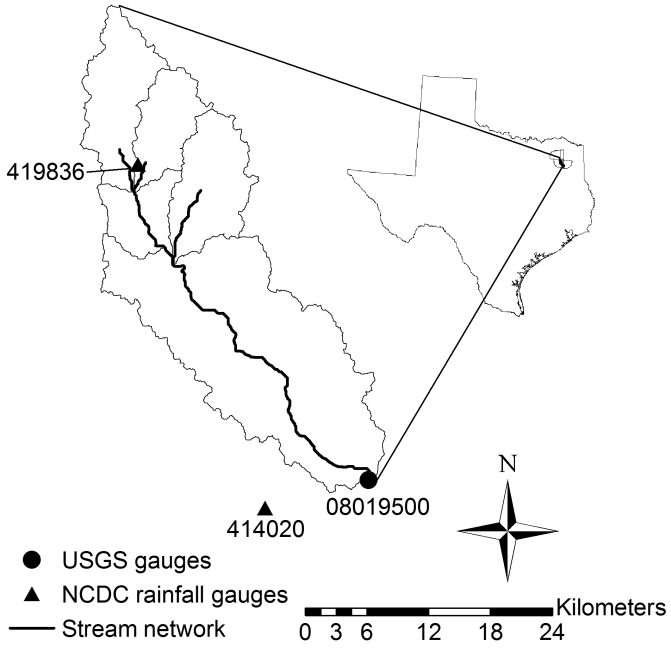


Figure 19: Big Sandy Creek watershed.

condition (Olivera et al., 2006), a one-year initial stabilization period was appended in front of each eight-month simulation period; the output for these stabilization periods was not used for model evaluation.

5.3.2. Calibration of the model parameters

The Soil and Water Assessment Tool (SWAT) is a long-term hydrologic model capable of simulating streamflows and sediment discharges on a daily basis (Arnold et al., 1998). SWAT subdivides a watershed into a number of subwatersheds, each of which has a main channel and tributary channels. Main channels are dendritically connected to each other. Each subwatershed consists of unique combinations of land use and soil type referred to as hydrologic response units (HRUs). Runoff and suspended sediment generated from the HRUs are routed through the stream network to the watershed outlet.

The Natural Resources Conservation Service (NRCS) curve number method (Soil Conservation Service, 1972) was used to calculate the runoff volume, and the variable storage routing method (Williams, 1969) was used to route streamflow. Sediment discharge generated with the Modified Universal Soil Loss Equation (MUSLE) (Williams, 1995) was routed using a simplified version of the Bagnold stream power equation (Bagnold, 1977). The model parameters listed in Table 16 were adjusted in calibration. There are three levels of the model parameters in SWAT as follows: the watershed-level parameters have one parameter value for the watershed, the subwatershed-level parameters per subwatershed, and the HRU-level parameters per HRU. In order to reduce the number of the model parameters to calibrate and keep the relative relationship between spatially distributed parameter values, the following one-parameter rule was employed to modify the parameter values:

$$p_{\text{new}} = p_0 + \alpha|p_{\text{b}} - p_0| \tag{5.3}$$

where p_{new} is the new parameter value, p_0 is the initial parameter value, p_{b} is either the upper or lower boundary of the parameter value, and α is a real number ranging from -1.0 to 1.0 and corresponds to one parameter. p_{b} was defined as

$$p_{\text{b}} = 0.5p_{\text{u}}(1 + \operatorname{sgn}\alpha) + 0.5p_{\text{l}}(1 - \operatorname{sgn}\alpha) \tag{5.4}$$

where p_{u} and p_{l} are the upper and lower boundary values of the parameter, respectively. For the Manning's n values for the tributary and main channels (CH_N(1) and CH_N(2), respectively, in the SWAT documentation), one α variable was used to keep the relative relationship between the tributary and main channels; the same rule was applied to the effective hydraulic conductivities in the tributary and main channel alluviums (CH_K(1) and CH_K(2), respectively, in the SWAT documentation). Therefore, there are 19 model parameters but 17 α variables (i.e., 17 dimensions).

Non-informative prior likelihood measure distributions (i.e., uniform distributions for $L_{\text{prior}}(\theta)$ over $\alpha \in [-1, 1]$, where θ is a set of p_{new} values) were used for all the α variables.

The objective function for streamflow was defined as the sum of the squares of the residuals (SSR):

$$f_Q = \sum_{i=1}^{n} \left(Q_i - \hat{Q}_i \right)^2 \tag{5.5}$$

where Q_i and $\hat{Q}_i$ are the observed and simulated streamflows at the i^{th} day, respectively, and n is the number of simulation days. Similarly, the objective function for sediment discharge was defined as

$$f_S = \sum_{i=1}^{n} \left(S_i - \hat{S}_i \right)^2 \tag{5.6}$$

where S_i and $\hat{S}_i$ are the observed and simulated sediment discharges at the i^{th} day, respectively. Eqs. (5.5) and (5.6) were normalized and aggregated into a single global optimization criterion (GOC) according to the Euclidean distance to the perfect model at the origin:

$$\text{GOC} = \left\{ \left[\frac{f_Q}{\sum_{i=1}^{n} \left(Q_i - \bar{Q} \right)^2} \right]^2 + \left[\frac{f_S}{\sum_{i=1}^{n} \left(S_i - \bar{S} \right)^2} \right]^2 \right\}^{1/2} \tag{5.7}$$

where $\bar{Q}$ is the mean observed streamflow, and $\bar{S}$ is the mean observed sediment discharge. The calibration of the model parameters was performed using GOC as the objective function.

Because, in this study, an optimization run is performed to take preferable parameter samples (i.e., behavioral models) which are then used for the GLUE uncertainty analysis, optimal models found by optimization must have an optimal likelihood measure. Therefore, the likelihood measure of the model parameters, $L(\theta|\xi, y)$, needs to be carefully defined to reflect well the objective function surface (i.e., the GOC func-

Table 16: List of the model parameters for calibration of streamflow and sediment discharge. The descriptions were taken from Neitsch et al. (2002b).

Parameter	Description	Range
CN2	Initial NRCS runoff curve number for moisture condition II	35–99
SOL_AWC	Available water capacity of the soil layer (mm H_2O/mm soil)	0.0–1.0
ESCO	Soil evaporation compensation factor	0.01–1.0
GWQMN	Threshold depth of water in the shallow aquifer required for return flow to occur (mm H_2O)	0–5000
GW_REVAP	Groundwater revap coefficient	0.02–0.2
REVAPMN	Threshold depth of water in the shallow aquifer for revap or percolation to the deep aquifer to occur (mm H_2O)	0–500
OV_N	Manning's n value for overland flow	0.01–1.0
CH_N(1)	Manning's n value for the tributary channels	0.01–0.5
CH_N(2)	Manning's n value for the main channel	0.01–0.5
CH_K(1)	Effective hydraulic conductivity in tributary channel alluvium (mm/hr)	0.025–250
CH_K(2)	Effective hydraulic conductivity in main channel alluvium (mm/hr)	0.025–250
ALPHA_BF	Baseflow alpha factor (days)	0.0–1.0
USLE_P	USLE equation support practice factor	0.25–1.0
APM	Peak rate adjustment factor for sediment routing in the tributary channels	1.0–2.0
PRF	Peak rate adjustment factor for sediment routing in the main channel	1.0–2.0
SPCON	Linear parameter for calculating the maximum amount of sediment that can be reentrained during channel sediment routing	0.0001–0.01
SPEXP	Exponent parameter for calculating sediment reentrained in channel sediment routing	1.0–2.0
CH_COV	Channel cover factor	0.0–1.0
CH_EROD	Channel erodibility factor	0.0–1.0

tion surface). The Nash-Sutcliffe coefficient (NS) (Nash and Sutcliffe, 1970) has been widely used to evaluate hydrologic models (Moriasi et al., 2007) and is defined as

$$\mathrm{NS} = 1 - \frac{\sum_{i=1}^{n} \left(X_i - \hat{X}_i \right)^2}{\sum_{i=1}^{n} \left(X_i - \bar{X} \right)^2} \tag{5.8}$$

where X_i and $\hat{X}_i$ represent the observed and simulated values of the variable of interest at the i^{th} time step, respectively, and $\bar{X}$ is the mean observed value of the variable. $\bar{X}$ is referred to as the "no-model" because this value is calculated without using any hydrologic concepts. Eq. (5.7) can be rewritten using the NS coefficient as follows:

$$\mathrm{GOC} = \left[(1 - \mathrm{NS}_Q)^2 + (1 - \mathrm{NS}_S)^2 \right]^{1/2} \tag{5.9}$$

where NS_Q and NS_S are the NS coefficients for streamflow and sediment discharge, respectively. When both NS coefficients are 0, GOC becomes $\sqrt{2}$ which is defined as non-behavioral in this study; when they are both 1, GOC becomes 0 which indicates the perfect model. This characteristic was used to define the likelihood measure as follows:

$$L(\theta|\xi, y) = \max\left(1 - \frac{\mathrm{GOC}}{\sqrt{2}}, 0\right). \tag{5.10}$$

Note that GOC is the distance from the origin (i.e., the perfect model) to a point measuring the model performance in the normalized objective function space (i.e., the normalized f_Q-f_S space), and a set of the model parameters is rejected as non-behavioral if the distance is greater than or equal to $\sqrt{2}$. Because the likelihood measure is a shifted and scaled-down version of GOC, optimal models found in the normalized f_Q-f_S space have an optimal likelihood measure. Likelihood measures greater than 0 do not imply that the NS coefficients for streamflow and sediment discharge are both greater than 0. Because the NS coefficient has limitations in reliably evaluating the model performance due to its high sensitivity to the no-model

as described in Chapter II, a model is still considered behavioral when GOC is less than $\sqrt{2}$ even if one of the NS coefficients is less than 0. Note that the likelihood measure defined in this study is not a probabilistic measure of how closely a model agrees with actual data, but the author's degree of subjective belief about how good the model is according to the likelihood measure defined in Eq. (5.10). The author expects that a model is usable or behavioral when either NS_Q or NS_S is greater than 0, and the other NS coefficient is not too bad such that the likelihood measure is greater than 0 (i.e., the threshold value for behavioral models in this study).

The calibration period was used for an optimization run with ISPSO to take preferable samples from the search space. Because of the high-dimensional and multi-modal nature of the SWAT model, the number of particles (i.e., swarm size) in ISPSO was set to 46, 28 more particles than the recommended swarm size for 17-dimensional single-modal problems (i.e., $10 + \lfloor 2\sqrt{17} \rfloor = 18$). The number of iterations was set to 1,000 due to the limitation of computational resources. Therefore, a total SWAT runs of 46,000 were performed in calibration, and the 46,000 sets of the model parameters are referred to as samples in the ISPSO-GLUE approach.

The likelihood measure defined in Eq. (5.10) was used to perform uncertainty analysis using the GLUE approach. Random sampling was performed to take 46,000 samples (i.e., the same sample size as in the ISPSO-GLUE approach) from the search space; the random samples were used to estimate the GLUE uncertainty bounds and compare them with the ISPSO-GLUE uncertainty bounds. The following section discusses the results of the ISPSO-GLUE approach unless otherwise noted.

5.3.3. Results and discussion

After calibration, one local solution was found, and one best solution was determined. However, the best solution found in calibration is not the global optimum because

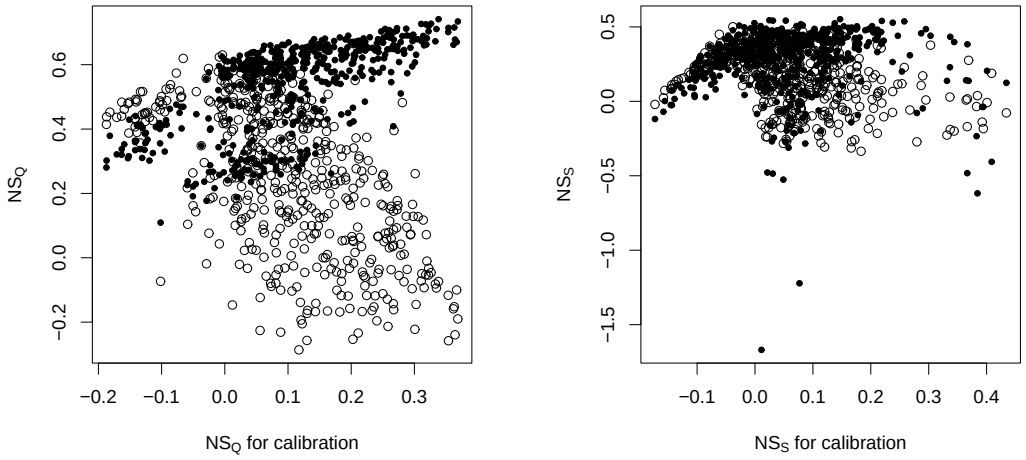
particles in the swarm did not converge to that point. The NS coefficients of the best and local solutions are shown in Table 17; the two solutions did not perform consistently well in the different simulation periods. One reason could be that the calibration period is too short for an optimization algorithm to calibrate the model parameters properly. Another reason may be that the SWAT model did not simulate one year of four seasonal processes in each simulation period. Because one simulation period partly overlaps with part of the other simulation periods in terms of seasons, the three periods show different characteristics of streamflow and sediment discharge. For this reason, optimal solutions in one period may not be able to perform well in the other periods. However, these results show that the model parameters are very sensitive to simulation periods, and, hence, they cannot be assumed to be time-invariant.

Table 17: Nash-Sutcliffe coefficients for the optimal solutions found in calibration.

Parameter set	Calibration period		Rejection period		Verification period	
	NS_Q	NS_S	NS_Q	NS_S	NS_Q	NS_S
Best solution	0.52	0.54	−1.23	−3.96	0.70	−1.07
Local solution	0.48	0.50	−1.38	−5.83	0.67	−2.48

If we are interested only in streamflow and treat the rejection and verification periods as validation and prediction periods, respectively, the two parameter sets may look reasonably good after calibration. However, we may not want to trust these parameter sets anymore after validation and would not use them for prediction, where they actually performed even better than in calibration. If the rejection and verification periods are treated as prediction and validation periods, respectively, we may be satisfied with the performances of the solutions because of the high NS coefficients in validation, but will fail to reliably predict unobserved streamflows in prediction.

These two scenarios clearly show the weakness of pursuing a single optimal parameter set without accounting for uncertainty. This property of optimal solutions can be seen in Figure 20 which shows the relationship between NS coefficients for the different simulation periods. Better calibrated models did not necessarily provide better performances in rejection, and mathematically inferior models performed more consistently, with a lower performance, than superior models especially in the prediction of streamflow. Note that the two clouds of circles and dots overlap near the region of NS_Q from 0 to 0.2, which suggests that those samples are more robust in terms of performance consistency.



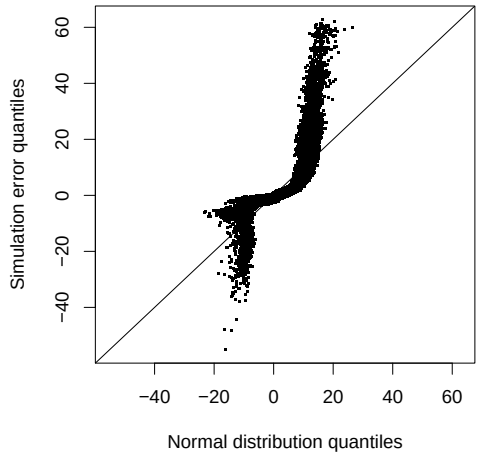
(a) Streamflow (b) Sediment discharge

Figure 20: Scatterplots of Nash-Sutcliffe coefficients. The circles and the dots indicate NS coefficients for rejection and verification, respectively.

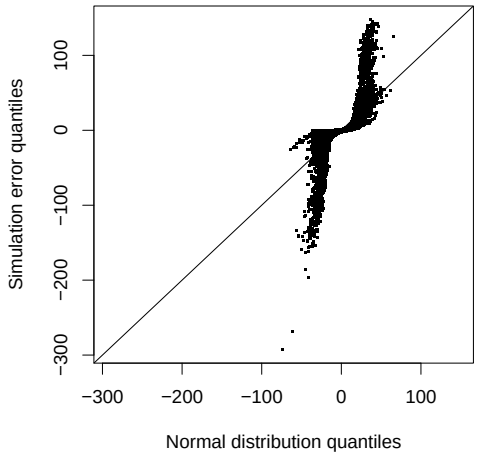
After calibration, 9,622 model parameter sets out of 46,000 had a likelihood measure greater than 0 and were considered behavioral models after calibration. Among the 9,622 behavioral models after calibration, 499 models had a likelihood measure greater than 0 after rejection and were used for verification. This study does not assume statistical characteristics of the model errors such as normal or log-normal

distributions. The normal quantile-quantile (Q-Q) plots in Figures 21a and 21b show that the assumption of normal distributions for the model errors is not appropriate, at least, for this study. As shown in Figures 21c and 21d, the log transformation of the model errors may justify the assumption of log-normal distributions for the model errors. However, the applicability of this assumption may be also limited because error variances are not constant. Of course, there may be other statistical models appropriate for this case study; however, since the main goal of this study is to reduce the computational burden of the GLUE framework which is not based on statistical assumptions, no further investigation was performed about the selection of statistical error models.

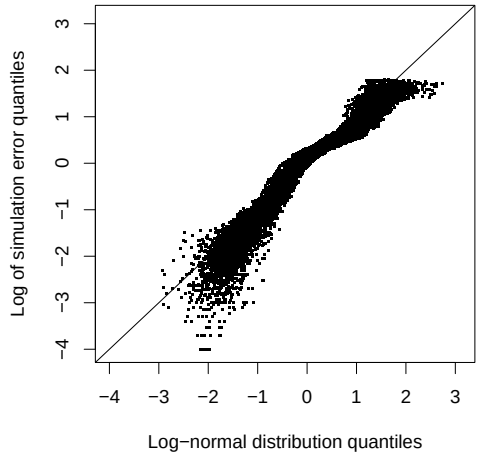
The α coordinates for the curve number and the USLE practice factor (CN2 and USLE_P in the SWAT documentation, respectively) of the best and local solutions found in calibration are (0.545, 0.279) and (0.532, 0.085), respectively. Figure 22 shows the cumulative marginal likelihood measure distributions of the α values for the curve number and the USLE practice factor. Non-informative prior likelihood measure distributions (i.e., uniform distributions) were updated after the calibration data became available. The posterior likelihood measure distributions after calibration were further updated with the rejection data. The cumulative marginal likelihood measure distribution of the α value for the curve number was significantly shifted to the left after the rejection data became available as shown in Figure 22a. This shift implies that the most likely α value (i.e., the point with the steepest slope in the cumulative distribution plot) for the curve number after calibration (e.g., approximately 0.5–0.6) is not the most likely after rejection. Among others listed in Table 16, the SWAT parameters such as ESCO, GWQMN, ALPHA_BF, PRF, SPCON, and CH_COV show a similar behavior. In contrast, for the other parameters such as the USLE practice factor, the marginal likelihood measure distribution was not sensitive



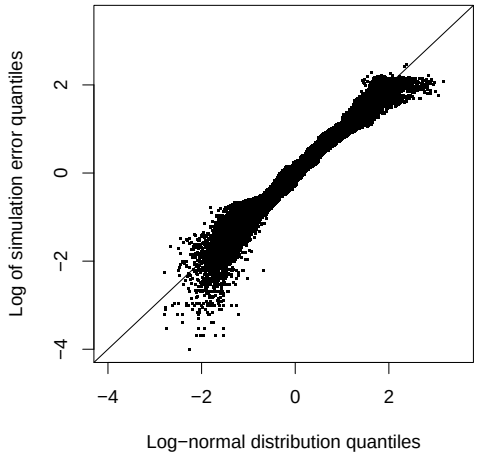
(a) Normal Q-Q plot for $Q_i - \hat{Q}_i$.



(b) Normal Q-Q plot for $S_i - \hat{S}_i$.



(c) Log-normal Q-Q plot for $|Q_i - \hat{Q}_i|$.



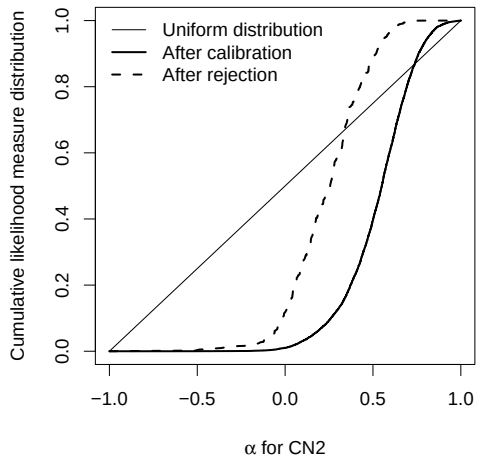
(d) Log-normal Q-Q plot for $|S_i - \hat{S}_i|$.

Figure 21: Normal Q-Q plots and log-normal Q-Q plots for the simulation errors of all behavioral models after rejection. The 1:1 lines indicate the perfect match. Since there are 499 behavioral models after rejection, each figure shows 499 independent Q-Q plots.

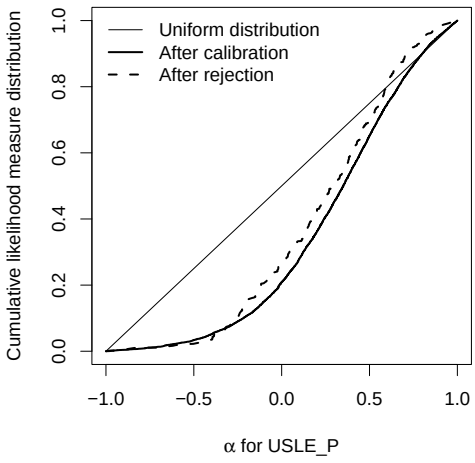
to the data used for calibration and rejection, and their most likely values did not change significantly over calibration and rejection. The difference in the sensitivities of different model parameters to the observed data implies that, for this particular dataset, a single parameter set cannot represent the watershed over different periods.

Using the GLUE approach, 1,115 models were classified as behavioral among 46,000 random samples when the calibration data became available, and 105 models survived rejection. These two sets of the behavioral models were used to build the cumulative likelihood measure distributions of the α values for the curve number and the USLE practice factor shown in Figures 22c and 22d. As shown in Figures 22a and 22c, the GLUE approach does not give as much weight to the most likely α value for the curve number (e.g., approximately 0.5–0.6) as the ISPSO-GLUE approach does because the random samples in the GLUE approach do not cluster around local optima. For the USLE practice factor, the cumulative likelihood measure distribution obtained using the GLUE approach (Figure 22d) did not change significantly from the uniform distribution, which implies that the random samples could not identify local optima. In contrast, the ISPSO-GLUE approach found many behavioral models close to local optima (e.g., approximately 0.1–0.5) and updated the prior cumulative distribution significantly to obtain an S-shaped posterior cumulative distribution as shown in Figure 22b. After rejection, the GLUE approach further updated the posterior cumulative distribution and identified the highly likely region around 0.15–0.30; however, the cumulative distribution after rejection is not as smooth as that obtained using the ISPSO-GLUE approach due to the lack of samples.

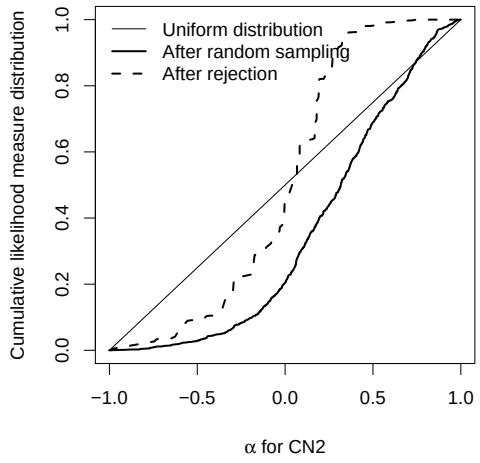
Figures 23 and 24 show the 95% uncertainty bounds in the verification period of the behavioral models obtained using the ISPSO-GLUE approach and the GLUE approach. Note that the 95% uncertainty bounds obtained using the two different approaches are qualitatively comparable when all of the 46,000 samples are taken into



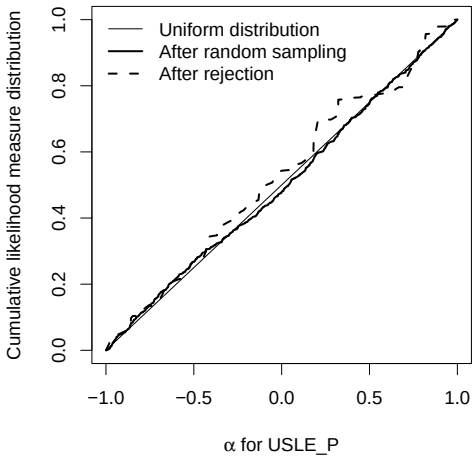
(a) CN2 using the ISPSO-GLUE approach



(b) USLE_P using the ISPSO-GLUE approach



(c) CN2 using the GLUE approach

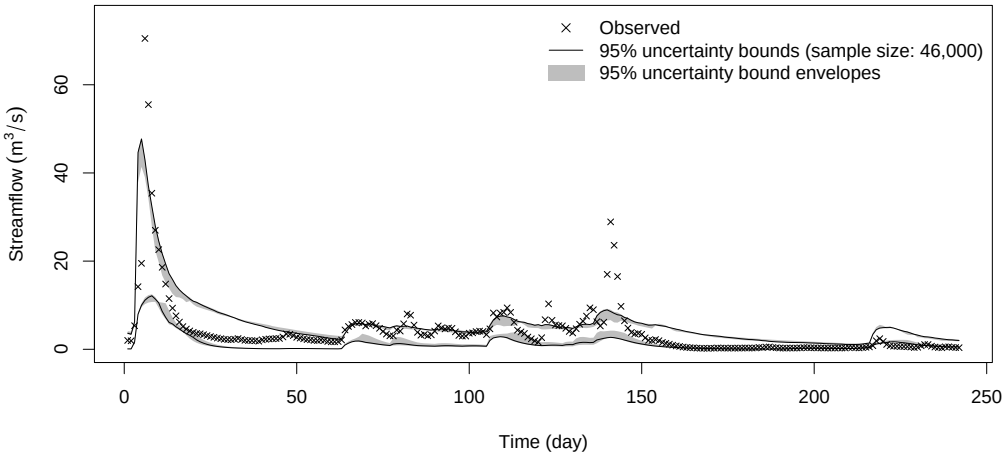


(d) USLE_P using the GLUE approach

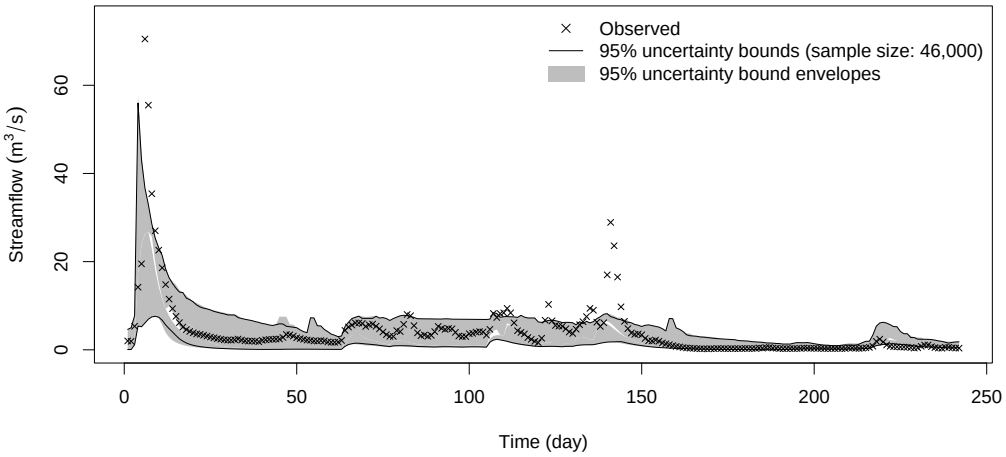
Figure 22: Cumulative marginal likelihood measure distributions of the α values for the curve number (CN2) and the USLE practice factor (USLE_P).

account except that the ISPSO-GLUE uncertainty bounds are slightly narrower than the GLUE uncertainty bounds. As shown in Figure 25, the ISPSO-GLUE approach underestimates the prediction uncertainty compared to the GLUE approach. This difference is due to the high frequency of highly likely parameter sets in the ISPSO-GLUE approach compared to the GLUE approach. Because of this difference, the ISPSO-GLUE uncertainty bounds miss a number of the peak observed values that the GLUE uncertainty bounds do not. The ISPSO-GLUE uncertainty bounds enclose 74% and 85% of the observed values for streamflow and sediment discharge, respectively, while the GLUE uncertainty bounds enclose 85% and 92% of them, respectively. These properties of the uncertainty bounds also show that the model errors of the behavioral models are not log-normally distributed, and their variances are not constant over time, which reinforces the discussion made about Figures 21c and 21d.

Since the SWAT model is computationally expensive, available computational resources are often very limited and, in many cases, one cannot run the model a large enough number of times until the uncertainty bounds are stabilized. The envelopes for the lower and upper limits of the uncertainty bounds in Figures 23 and 24 show how sensitive the uncertainty bounds are to the sample size. The 46,000 samples from each approach were numbered from the first sample to the last one; samples 1 to $1,000k$ $\forall k = 1, \cdots, 46$ were used to build uncertainty bounds if there were more than one behavioral model after rejection. For example, in the GLUE approach, first 1,000 samples included only one behavioral model after rejection and could not build uncertainty bounds; first 2,000 samples included three behavioral models after rejection and were used to estimate uncertainty bounds. This test simulates the two approaches with 46 different sample sizes; the much narrower gray regions in Figures 23a and 24a compared to Figures 23b and 24b show that the convergence

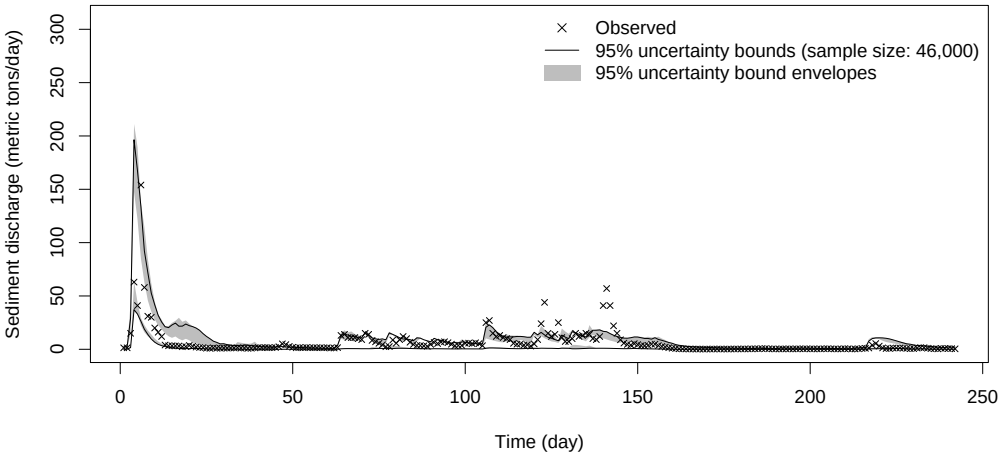


(a) Streamflow using the ISPSO-GLUE approach.

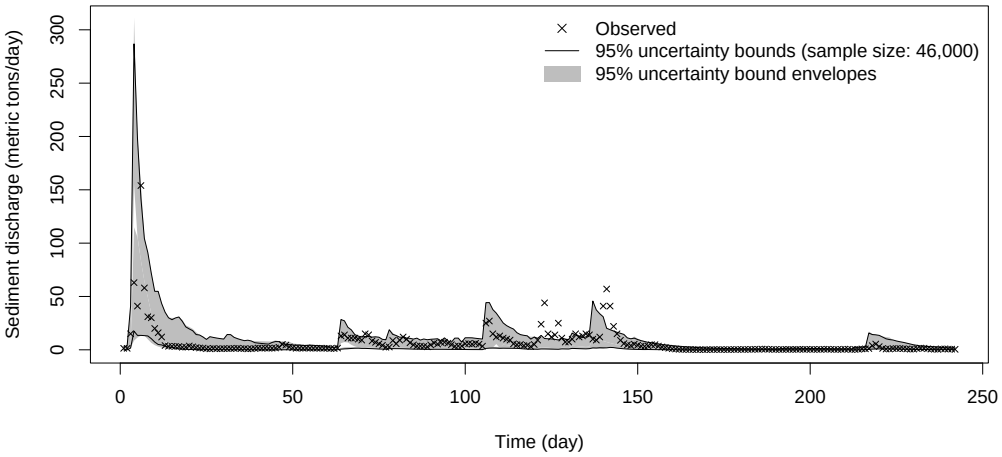


(b) Streamflow using the GLUE approach.

Figure 23: 95% ISPSO-GLUE uncertainty bounds and 95% GLUE uncertainty bounds of streamflow in the verification period. The gray regions denote the envelopes for the lower and upper limits of the uncertainty bounds estimated using the sample size of 1,000 to 46,000 at 1,000 intervals; the uncertainty bounds were not evaluated for samples including only one behavioral model. Each of the gray regions encompasses 46 and 45 upper or lower limits of the uncertainty bounds for the ISPSO-GLUE approach and the GLUE approach, respectively.

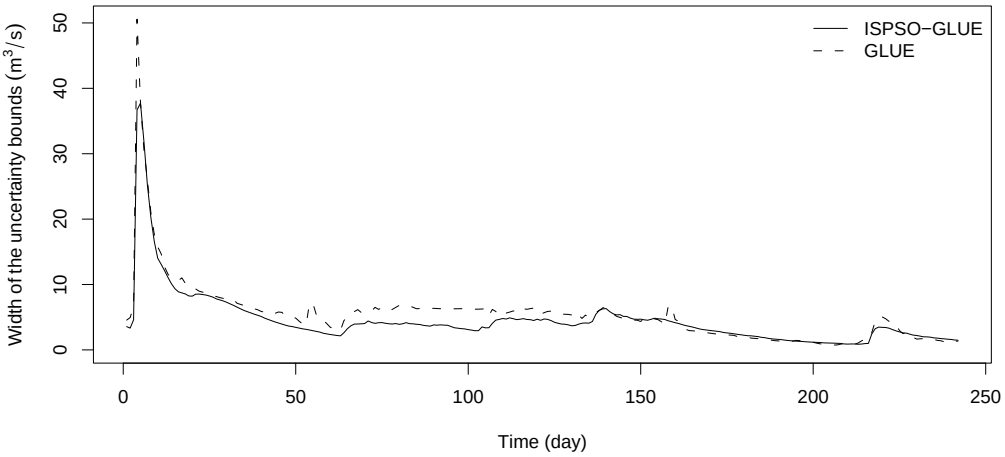


(a) Sediment discharge using the ISPSO-GLUE approach.

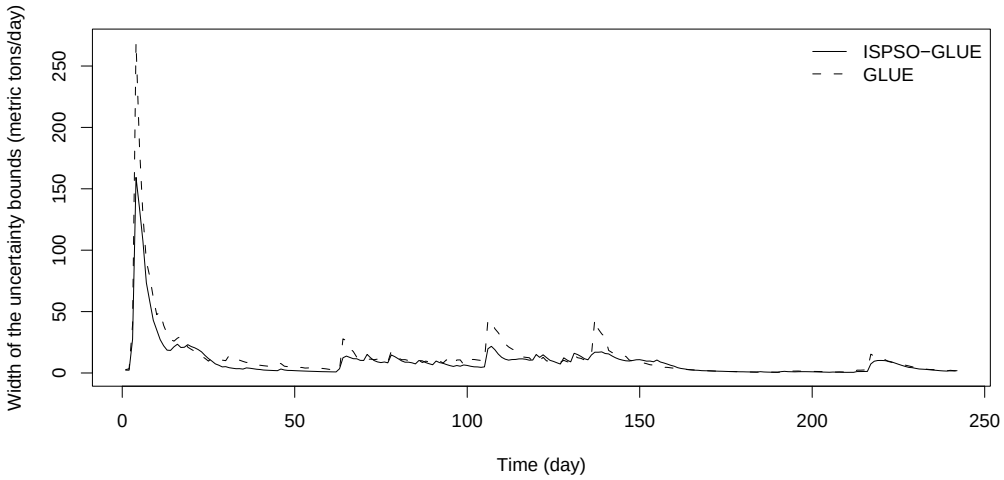


(b) Sediment discharge using the GLUE approach.

Figure 24: 95% ISPSO-GLUE uncertainty bounds and 95% GLUE uncertainty bounds of sediment discharge in the verification period. The gray regions denote the envelopes for the lower and upper limits of the uncertainty bounds estimated using the sample size of 1,000 to 46,000 at 1,000 intervals; the uncertainty bounds were not evaluated for samples including only one behavioral model. Each of the gray regions encompasses 46 and 45 upper or lower limits of the uncertainty bounds for the ISPSO-GLUE approach and the GLUE approach, respectively.



(a) Width of the uncertainty bounds of streamflow.



(b) Width of the uncertainty bounds of sediment discharge.

Figure 25: Width of the uncertainty bounds for a sample size of 46,000.

rate of the ISPSO-GLUE uncertainty bounds is much faster than that of the GLUE uncertainty bounds. In fact, the ISPSO-GLUE uncertainty bounds of 2,000 samples are close enough to those of 46,000 samples while the GLUE uncertainty bounds of up to 14,000 samples are not yet stabilized. The white regions surrounded by the lower and upper envelopes are "the guaranteed uncertainty range" when an arbitrary number of samples from 1,000 to 46,000 are taken. The guaranteed uncertainty range of the GLUE approach is too narrow to locate any observed values within it while that of the ISPSO-GLUE approach is similar to its uncertainty bounds and encloses 52% and 60% of the observed values for streamflow and sediment discharge, respectively. Figure 26 shows how the likelihood measure of the median estimates of the uncertainty bounds changes with different sample sizes. The ISPSO-GLUE approach consistently outperformed the GLUE approach regardless of the sample size. The average likelihood measure is 0.53 for the ISPSO-GLUE approach and 0.44 for the GLUE approach. On average, the likelihood measure was improved by 20% by using the ISPSO-GLUE approach. Moreover, the ISPSO-GLUE approach shows a quicker convergence to the final state compared to the GLUE approach. These results imply that choosing an appropriate sample size is more critical for the GLUE approach, and the better exploration of the search space is needed for the ISPSO-GLUE approach to widen the uncertainty bounds by taking less likely yet behavioral parameter sets.

The failure of capturing the observed variables within the uncertainty bounds over the whole verification period suggests the followings: (1) the rejection period is so much different from the calibration period that good models barely survive rejection that, otherwise, would provide good prediction in verification, (2) 46,000 samples may not be enough to well characterize the likelihood measure surface, (3) the one-parameter (α) rule to update spatially distributed parameter values may be in-

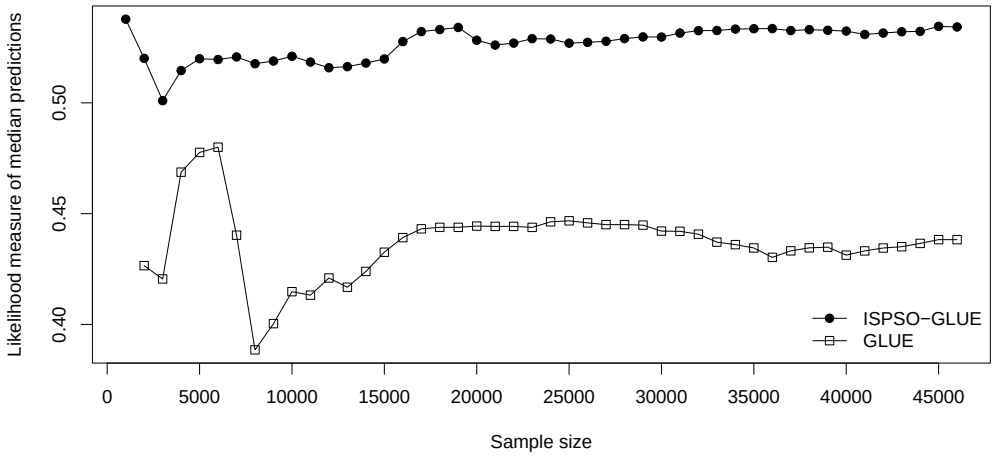


Figure 26: Likelihood measure of median predictions versus sample size. Note that, with a sample size of 1,000, the GLUE approach found only one behavioral model, and no median predictions were available.

appropriate to properly take into account the spatial variability of the data, (4) there might be errors in the observed data or the model structure that prevent providing consistent prediction over the verification period, and (5) it could be the subjective likelihood measure used in this study or the GLUE approach itself.

The first argument may be valid; however, as already discussed, in real prediction, we never know whether or not the prediction period is similar enough to the calibration period for which the model parameters are calibrated. For this reason, the behavioral models need to be tested with much different datasets; rejection does this role in this study.

The second argument is probably right because, even after random sampling of 46,000 sets of the model parameters, only 105 behavioral models could be used to build the uncertainty bounds in the case of the GLUE approach. We know that taking more samples is always not enough when it comes to characterizing the likelihood

measure surface in such high-dimensional problems (e.g., 17-dimensional problem in this study); however, we cannot ignore the time taken to perform this type of uncertainty analysis. It takes around 16 days (i.e., 384 hours) to run the SWAT model of the Big Sandy Creek watershed 46,000 times with random sampling using R 2.5.1 (R Development Core Team, 2006) on a Dell(R) Optiplex(TM) GX620 computer (Pentium(R) D CPU 2.80 GHz with 3 GB of RAM) operated by MS-Windows(TM) XP. Figure 27 shows the relationship between the number of behavioral models after rejection and the sample size; the two curves show almost the linear relationship between the two variables although it is not ensured that the curves will be still linear even after a huge amount of sampling. Using the GLUE approach, to obtain the same number of behavioral models as in the ISPSO-GLUE approach, it will take approximately 76 days (i.e., $499/105 \times 16$ days) assuming that the linear relationship in Figure 27 will be still valid until then. However, it does not seem practical to spend over two months for just one uncertainty analysis. Even worse is that, if the number of subwatersheds significantly increases from only five as in this study, the time requirement will become even more demanding. Of course, parallel computing or supercomputing may help a lot in this regard at the expense of financial resources. The current trend of the development of powerful computational devices and facilities is promising for the GLUE framework as anticipated by Beven (2006a).

As for the third argument, the α function defined in Eq. (5.3) may not be appropriate for the SWAT model to take into account the spatial variability of the data; however, the spatial variability in the Big Sandy Creek watershed may not be an important factor in uncertainty analysis as Chapter II shows that, in watersheds whose time of concentration is shorter than the computational time step of the SWAT model, the model output is not sensitive to the spatial variability of land use, soil type, and precipitation data. This phenomenon is because everything happens within

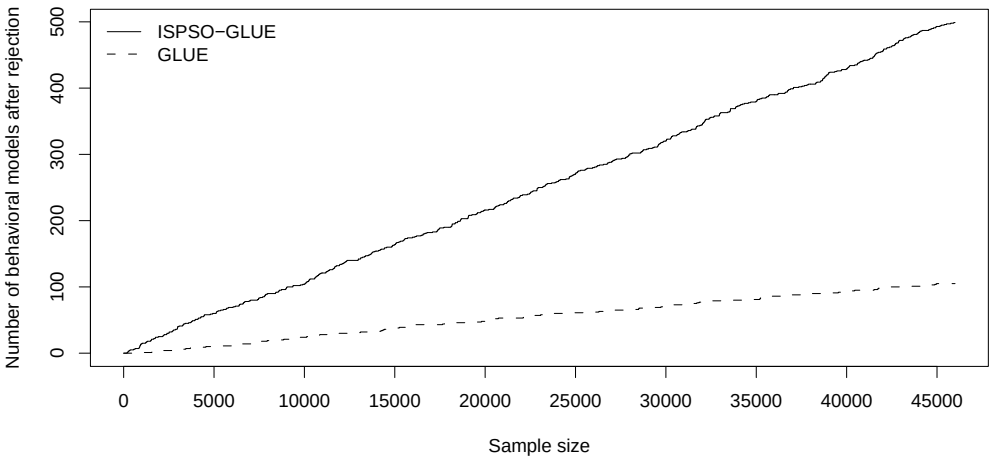


Figure 27: Number of behavioral models after rejection versus sample size.

the watershed in a single time step, and the model is not able to simulate such details within one computational time step. Because the area of the Big Sandy Creek watershed is 598 km^2, which is smaller than two of the three study areas used in Chapter II (e.g., 831-km^2 and 1,005-km^2 watersheds in Texas), it is unlikely that the uncertainty of the model output is significantly affected by how spatially distributed parameter values are updated.

There might be the measurement error or the model structural error. As for the measurement error, accumulated rainfall was measured every day at 07:00 and 08:00 at gauges 414020 and 419836, respectively, while daily observed values of streamflow and sediment discharge were processed appropriately after measurement. This mismatch in the time of observation may lead to poor model performances; however, because over 70% of the observed values are enclosed by the uncertainty bounds, this may not be the sole problem, at least, in this study. Another possibility might be that, due to the model structural error, not many sets of the model parameters could produce good predictions describing well the observed values of both streamflow and sediment

discharge over the whole verification period and the uncertainty bounds missed a number of the observed values.

Lastly, the likelihood measure used in this study that evaluates simulated streamflows and sediment discharges at the same time may be too rigorous to find an enough number of behavioral models, and using less strict likelihood measures may help widen the uncertainty bounds to enclose all the observed values. The results of this study may be revealing the weakness of the GLUE approach; however, as evidenced earlier, a simple statistical assumption about the model errors could not be well justified for this study, which undermines the basis of the probabilistic Bayesian framework and suggests that careful consideration should be given to the selection of statistical error models.

The model structure cannot be perfect inherently, and it is not possible to find a single optimal solution that provides consistent predictions with different observations. Similarly, even multiple behavioral models may not be able to guarantee consistent predictions. These results suggest the followings: (1) we cannot completely rely on a single set of the model parameters even if it was well calibrated and (2) in order to better ensure consistent predictions, even sub-optimal regions of the search space need to be investigated. The ultimate goal of the model calibration is not to match observed data, but to predict unobserved variables. In this perspective, it is essential to avoid over-trusting few mathematically optimal solutions and evaluate the usefulness of sub-optimal ones to find out the strengths and the weaknesses of the model structure. This process, of course, may require a significant number of model runs, which usually leads to a high computational burden. Further, in high-dimensional problems, sub-optimal regions of the search space may not be concentrated near only the global optimum, and better strategies are needed than global optimization algorithms to search for multiple behavioral models as well as to

avoid sampling from the inferior regions of low likelihood measures to significantly reduce model runs. To this end, a multi-model optimization algorithm called ISPSO was employed not only to find optimal solutions in a high-dimensional search space, but also to perform uncertainty analysis in the GLUE framework.

The ISPSO-GLUE uncertainty bounds inherently underestimate the prediction uncertainty compared to the GLUE uncertainty bounds because the ISPSO-GLUE approach gives much weight to highly likely models compared to the GLUE approach; however, the two uncertainty bounds were not significantly different. Given the similar results, the convergence rate of the uncertainty bounds is an important factor for uncertainty analysis especially when the model run is computationally very expensive. The ISPSO-GLUE uncertainty bounds were not significantly sensitive to the sample size, and their convergence to the uncertainty bounds of a higher sample size (i.e., 46,000 in this study) was much faster than that of the GLUE uncertainty bounds. This fast convergence of the uncertainty bounds may be a promising feature of the ISPSO-GLUE approach when uncertainty analysis is performed for costly distributed hydrologic models.

5.4. Summary and conclusions

This chapter discussed the feasibility of the application of a multi-modal optimization algorithm called ISPSO to uncertainty analysis within the GLUE framework. The SWAT model was used to estimate daily streamflows and sediment discharges at the watershed outlet, and one multi-modal optimization run was performed. Two mathematically optimal solutions did not show consistent model performances over different simulation periods, and even inferior models provided more consistent performances. The time-variant nature of a model parameter was discussed by building

the cumulative marginal likelihood measure distribution of the model parameter. Because any model is a combination of different model parameters, no single model can be time-invariant and thus be the only good realization of the real system. Samples taken in optimization were used to estimate the prediction limits of the behavioral models called "the ISPSO-GLUE uncertainty bounds." The typical GLUE approach was also performed with random sampling of the same sample size to obtain "the GLUE uncertainty bounds." The two uncertainty bounds were qualitatively comparable except that the ISPSO-GLUE uncertainty bounds slightly underestimated the prediction uncertainty compared to the GLUE uncertainty bounds and thus missed a number of the observed values; however, the convergence rate of the ISPSO-GLUE uncertainty bounds was much faster than that of the GLUE uncertainty bounds. This feature of the proposed approach allows us to obtain the uncertainty bounds much faster at the expense of missing a number of the observed values especially when a model run is computationally very expensive.

Unlike random sampling techniques usually employed for the GLUE approach, ISPSO not only takes samples uniformly, when it is allowed, but also tries to find optimal solutions. This unique feature of ISPSO increases the chance of discovering behavioral models widely spread in high-dimensional search spaces and provides a number of uniform samples for the purpose of uncertainty analysis. In addition, optimal solutions found by the algorithm can be used to suggest sets of the model parameters instead of several hundred behavioral models for the purpose of quick prediction. However, the time-variant nature of some model parameters suggests that great care should be taken when prediction is made using mathematically optimal solutions, and the model structure might need to be refined so that it only parameterizes time-invariant or data-insensitive characteristics of the system. A comparison between the ISPSO-GLUE approach and the GLUE approach suggests that the ISPSO-GLUE

approach may be a good alternative to the computationally more expensive GLUE approach.

CHAPTER VI

SUMMARY AND CONCLUSIONS

Chapter II discussed the effect of the spatial distribution of land use, soil type, and rainfall on estimated streamflows in watersheds with times of concentration shorter than the computational time step of the SWAT model. The model was not able to identify differences in substantially different configurations of the spatial data, and even the use of uniform land use and soil type maps was not noticeable. When the watershed is small compared to the computational time step of the model, the model is not necessarily able to identify realistic representations of the spatial data and provide better model performances with them. However, spatially distributed data help understand the characteristics of the watershed and provide distributed hydrologic models valuable information. A special care should be taken in generalization of the results in this chapter because watersheds where high correlations exist between land use, soil type, and rainfall data might require accounting for the spatial variability to estimate more accurate runoff volumes.

In Chapter III, a population-based multi-modal optimization algorithm called the Isolated Speciation-based Particle Swarm Optimization (ISPSO) was developed. Parameter samples are referred to as particles in particle swarm optimization, and they form species in ISPSO based on Euclidean proximity. Isolated particles left alone after the proximity-based speciation form an additional species called "the isolated species" to make themselves move around the search space. A deterministic low-discrepancy sampling technique called the Sobol' sequence was employed to enhance the diversity of particles. Seven mathematical functions were used to test the performance and the scalability of ISPSO, and the results were compared with the reported results of SPSO (Li, 2004) and NichePSO (Brits et al., 2007). ISPSO outperformed

SPSO and NichePSO in terms of computational cost, consistency, and scalability. The premature convergence of particles and computational complexity required at each iteration need to be improved in the future work.

In Chapter IV, the number of minima of the Griewank function, one of the test functions used in Chapter III, was analytically derived. It is difficult to directly count the number of minima because of the complicated nature of the function's surface and high correlations between dimensions. By redefining the problem of counting the number of minima as counting the number of tangent points lying on the parabolic plane, which is part of the function, it was possible to analytically count the number of minima. A numerical method was developed to find hyperrectangles in which the analytical derivation of the number of minima of the function is valid. The maximum extents of the hyperrectangles for up to four dimensions were estimated, and a reference table was provided for the numbers of minima for three different search spaces.

In Chapter V, ISPSO was applied to the GLUE framework to estimate the prediction uncertainty of the SWAT model; the proposed approach is referred to as "the ISPSO-GLUE approach" contrast to "the GLUE approach," which typically requires extensive random sampling. Optimal solutions found by ISPSO were evaluated to test whether or not a single optimal model could represent the watershed over different simulation periods. The optimal solutions were sensitive to the data used in optimization because of the time-variant nature of some model parameters; no single parameter set was able to show consistent model performance. Uncertainty analysis for streamflow and sediment discharge was performed by running one multi-modal optimization run, rejecting some of the "behavioral models" after calibration, and applying the behavioral models that survived rejection to verification. The uncertainty bounds obtained by this approach are referred to as "the ISPSO-GLUE uncertainty

bounds" while those obtained by the GLUE approach are referred to as "the GLUE uncertainty bounds." Random sampling of the same sample size as in the ISPSO-GLUE approach was performed to estimate the GLUE uncertainty bounds and compare them with those for the proposed approach. The ISPSO-GLUE uncertainty bounds slightly underestimated the prediction uncertainty compared to the GLUE uncertainty bounds; however, the convergence rate of the ISPSO-GLUE uncertainty bounds were much faster than that of the GLUE uncertainty bounds. The application of the ISPSO algorithm to the GLUE framework significantly reduces computational burden required to find many behavioral models in high-dimensional problems because the diversity of particles in ISPSO enhances the discovery of behavioral models compared to random samplings. This feature is promising especially when uncertainty analysis is performed for costly distributed hydrologic models. However, the bias of behavioral models found by ISPSO needs to be further investigated

There is a great deal of uncertainty in environmental modeling and especially in high-dimensional distributed hydrologic models. This dissertation addressed the issues of the uncertainty associated with the spatial variability of the data and the identification of the model parameters. The multi-modal optimization algorithm and the uncertainty analysis framework proposed in this dissertation will help reduce the predictive uncertainty.

REFERENCES

Acan, A., Gunay, A., 2005. Enhanced particle swarm optimization through external memory support. In: Proceedings of the Congress on Evolutionary Computation. Vol. 2. pp. 1875–1882.

American Society of Civil Engineers, 1993. Criteria for evaluation of watershed models. ASCE task committee on the definition of criteria for evaluation of watershed models of the watershed management committee. Journal of Irrigation and Drainage Engineering 119, 429–442.

Andréassian, V., Lerat, J., Loumagne, C., Mathevet, T., Michel, C., Oudin, L., Perrin, C., 2007. What is really undermining hydrologic science today? Hydrological Processes 21, 2819–2822.

Andréassian, V., Perrin, C., Michel, C., Usart-Sanchez, I., Lavabre, J., 2001. Impact of imperfect rainfall knowledge on the efficiency and the parameters of watershed models. Journal of Hydrology 250, 206–223.

Antonov, I. A., Saleev, V. M., 1979. An economic method of computing lp_τ sequences. USSR Computational Mathematics and Mathematical Physics 19 (1), 252–256.

Arnold, J. G., Srinivasan, R., Muttiah, R. S., Williams, J. R., 1998. Large area hydrologic modelling and assessment, Part I: Model development. Journal of the American Water Resources Association 34 (1), 73–89.

Aronica, G., Hankin, B., Beven, K., 1998. Uncertainty and equifinality in calibrating distributed roughness coefficients in a flood propagation model with limited data. Advances in Water Resources 22 (4), 349–365.

Bagnold, R. A., 1977. Bedload transport in natural rivers. Water Resources Research 13, 303–312.

Beasley, D., Bull, D. R., Martin, R. R., 1993. A sequential niche technique for multimodal function optimization. Evolutionary Computation 1 (2), 101–125.

Beven, K., 1993. Prophecy, reality and uncertainty in distributed hydrological modelling. Advances in Water Resources 16, 41–51.

Beven, K., 2006a. A manifesto for the equifinality thesis. Journal of Hydrology 320, 18–36.

Beven, K., 2006b. On undermining the science? Hydrological Processes 20, 3141–3146.

Beven, K., Binley, A., 1992. The future of distributed models: Model calibration and uncertainty prediction. Hydrological Processes 6, 279–298.

Beven, K., Freer, J., 2001. Equifinality, data assimilation, and uncertainty estimation in mechanistic modelling of complex environmental systems using the GLUE methology. Journal of Hydrology 249, 11–29.

Beven, K., Smith, P., Freer, J., 2007. Comment on "Hydrological forecasting uncertainty assessment: Incoherence of the GLUE methodology" by Pietro Mantovan and Ezio Todini. Journal of Hydrology 338, 315–318.

Beven, K., Young, P., 2003. Comment on "Bayesian recursive parameter estimation for hydrologic models" by M. Thiemann, M. Trosset, H. Gupta, and S. Sorooshian. Water Resources Research 39 (5), 1116.

Beven, K. J., Freer, J., Hankin, B., Schulz, K., 2000. The use of generalised likelihood measures for uncertainty estimation in high order models of environmental systems.

In: Fitzgerald, W. J., Smith, R. L., Walden, A. T., Young, P. C. (Eds.), Nonlinear and Nonstationary Signal Processing. Cambridge University Press, New York, New York, pp. 115–151.

Bingner, R. L., Garbrecht, J., Arnold, J. G., Srinivasan, R., 1997. Effect of watershed subdivision on simulation runoff and fine sediment yield. Transactions of the ASAE 40 (5), 1329–1335.

Blackwell, T., Branke, J., August 2006. Multiswarms, exclusion, and anti-convergence in dynamic environments. IEEE Transactions on Evolutionary Computation 10 (4), 459–472.

Brath, A., Montanari, A., Toth, E., 2004. Analysis of the effects of different scenarios of historical data availability on the calibration of a spatially-distributed hydrological model. Journal of Hydrology 291, 232–253.

Brill, Jr., E. D., May 1979. The use of optimization models in public-sector planning. Management Science 25, 413–422.

Brits, R., Engelbrecht, A. P., van den Bergh, F., 2002. A niching particle swarm optimizer. In: Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning. pp. 692–696.

Brits, R., Engelbrecht, A. P., van den Bergh, F., April 2003. Scalability of Niche PSO. In: Proceedings of the 2003 IEEE Swarm Intelligence Symposium. pp. 228–234.

Brits, R., Engelbrecht, A. P., van den Bergh, F., 2007. Locating multiple optima using particle swarm optimization. Applied Mathematics and Computation 189, 1859–1883.

Chaplot, V., 2005. Impact of DEM mesh size and soil map scale on SWAT runoff, sediment, and NO_3-N loads predictions. Journal of Hydrology 312, 207–222.

Chaplot, V., Saleh, A., Jaynes, D. B., 2005. Effect of the accuracy of spatial rainfall information on the modeling of water, sediment, and NO_3-N loads at the watershed level. Journal of Hydrology 312, 223–234.

Chelouah, R., Siarry, P., 2003. Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. European Journal of Operational Research 148, 335–348.

Chen, E., Mackay, D. S., 2004. Effects of distribution-based parameter aggregation on a spatially distributed agricultural nonpoint source pollution model. Journal of Hydrology 295, 211–224.

Clerc, M., 1999. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. In: Proceedings of the Congress on Evolutionary Computation. Vol. 3. pp. 1951–1957.

Clerc, M., September 2007. Standard PSO 2007. http://www.particleswarm.info/standard_pso_2007.c, accessed in November, 2007.

Clerc, M., Kennedy, J., February 2002. The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation 6 (1), 58–73.

Cotter, A. S., Chaubey, I., Costello, T. A., Soerens, T. S., Nelson, M. A., August 2003. Water quality model output uncertainty as affected by spatial resolution of input data. Journal of the American Water Resources Association 39 (4), 977–986.

Deb, K., 2004. Single and multi-objective optimization using evolutionary computation. In: Liong, S.-Y., Phoon, K.-K., Babovic, V. (Eds.), Proceedings of the 6th International Conference on Hydroinformatics. Vol. 1. World Scientific Publishing Co. Pte. Ltd., pp. 14–35.

Devonec, E., Barros, A. P., 2002. Exploring the transferability of a land-surface hydrology model. Journal of Hydrology 265, 258–282.

Di Luzio, M., Arnold, J. G., Srinivasan, R., 2005. Effect of GIS data quality on small watershed stream flow and sediment simulations. Hydrological Processes 19, 629–650.

Di Luzio, M., Srinivasan, R., Arnold, J. G., 2002. Integration of watershed tools and SWAT model into BASINS. Journal of the American Water Resources Association 38 (4), 1127–1141.

Dorigo, M., Maniezzo, V., Colorni, A., 1996. The ant system: Optimization by a colony of cooperting agents. IEEE Transactions on Systems, Man, and Cybernetics–Part B 26 (1), 1–13.

Draper, N. R., Box, G. E., January 1987. Empirical Model-Building and Response Surfaces. John Wiley and Sons Inc.

Duan, Q., Gupta, V. K., Sorooshian, S., 1993. Shuffled complex evolution approach for effective and efficient global minimization. Journal of Optimization Theory and Applications 76 (3), 501–521.

Duan, Q., Sorooshian, S., Gupta, V., 1992. Effective and efficient global optimization for conceptual rainfall-runoff models. Water Resources Research 28 (4), 1015–1031.

Duan, Q., Sorooshian, S., Gupta, V. K., 1994. Optimal use of the SCE-UA global optimization method for calibrating watershed models. Journal of Hydrology 158, 265–284.

Eberhart, R. C., Kennedy, J., 1995. A new optimizer using particle swarm theory. In: Proceedings of the 6th International Symposium on Micromachine and Human Science. Nagoya, Japan, pp. 39–43.

Eberhart, R. C., Shi, Y., 2000. Comparing inertia weights and constriction factors in particle swarm optimization. In: Proceedings of the Congress on Evolutionary Computation. Vol. 1. La Jolla, California, pp. 84–88.

Eberhart, R. C., Simpson, P., Dobbins, R., 1996. Computational Intelligence PC Tools. Academic, Ch. 6, pp. 212–226.

Eckhardt, K., Arnold, J. G., 2001. Automatic calibration of a distributed catchment model. Journal of Hydrology 251, 103–109.

Eckhardt, K., Breuer, L., Frede, H.-G., 2003. Parameter uncertainty and the significance of simulated land use change effects. Journal of Hydrology 273, 164–176.

Faure, H., 1982. Discrépance de suites associées à un système de numération (en dimension s). Acta Arithmetica 41, 337–351.

Faure, H., Tezuka, S., 2002. Another random scrambling of digital (t, s)-sequences. In: Fang, K. T., Hickernell, F. J., Niederreiter, H. (Eds.), Monte Carlo and Quasi-Monte Carlo Methods 2000. Springer-Verlag, Berlin, Germany, pp. 242–256.

Faurès, J.-M., Goodrich, D. C., Woolhiser, D. A., Sorooshian, S., 1995. Impact of small-scale spatial rainfall variability on runoff modeling. Journal of Hydrology 173, 309–326.

FitzHugh, T. W., Mackay, D. S., 2000. Impacts of input parameter spatial aggregation on an agricultural nonpoint source pollution model. Journal of Hydrology 236, 35–53.

Fontaine, T. A., Cruickshank, T. S., Arnold, J. G., Hotchkiss, R. H., 2002. Development of a snowfall-snowmelt routine for mountainous terrain for SWAT. Journal of Hydrology 262, 209–223.

Freer, J., Beven, K., Ambroise, B., July 1996. Byesian estimation of uncertainty in runoff prediction and the value of data: An application of the GLUE approach. Water Resources Research 32 (7), 2161–2173.

Galanti, S., Jung, A., 1997. Low-discrepancy sequences: Monte Carlo simulation of option prices. The Journal of Derivatives 5, 63–83.

Gan, T. Y., Biftu, G. F., 1996. Automatic calibration of conceptual rainfall-runoff models: Optimization algorithms, catchment conditions and model structure. Water Resources Research 32 (12), 3513–3524.

Gan, T. Y., Burges, S. J., 1990. An assessment of a conceptual rainfall-runoff model's ability to represent the dynamics of small hypothetical catchments, 1. Models, model properties, and experimental design. Water Resources Research 26 (7), 1595–1604.

Green, W. H., Ampt, G. A., 1911. Studies on soil physics, 1. The flow of air and water through soils. Journal of Agricultural Sciences 4, 11–24.

Griewank, A. O., May 1981. Generalized descent for global optimization. Journal of Optimization Theory and Applications 34 (1), 11–39.

Guanqi, G., Shouyi, Y., December 2003. Evolutionary parallel local search for function optimization. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics 33 (6), 864–876.

Gupta, H., Thiemann, M., Trosset, M., Sorooshian, S., 2003. Reply to comment by K. Beven and P. Young on "Bayesian recursive parameter estimation for hydrologic models". Water Resources Research 39 (5), 1117.

Gupta, H. V., Sorooshian, S., Yapo, P. O., 1998. Toward improved calibration of hydrologic models: Multiple and noncommensurable measures of information. Water Resources Research 34 (4), 751–763.

Hall, J., O'Connell, E., Ewen, J., 2007. On not undermining the science: Coherence, validation and expertise. Discussion of Invited Commentary by Keith Beven Hydrological Processes, 20, 3141–3146 (2006). Hydrological Processes 21, 985–988.

Halton, J. H., 1960. On the efficiency of certain quasi-random sequences of points in evaluating multi-dimensional integrals. Numerische Mathematik 2, 84–90.

Halton, J. H., Smith, G. B., 1964. Algorithm 247: Radical-inverse quasi-random point sequence. Communications of the ACM 7, 701–702.

Hanratty, M. P., Stefan, H. G., 1998. Simulating climate change effects in a Minnesota agricultural watershed. Journal of Environmental Quality 27, 1524–1532.

Hassan, A., 2004. Validation, proof-of-concept, and postaudit of the groundwater flow and transport model of the project Shoal area. US Department of Energy Publication No. 45206.

Haverkamp, S., Fohrer, N., Frede, H.-G., 2005. Assessment of the effect of land use patterns on hydrologic landscape functions: A comprehensive GIS-based tool to

minimize model uncertainty resulting from spatial aggregation. Hydrological Processes 19, 715–727.

He, S., Wu, Q. H., Wen, J. Y., Saunders, J. R., Paton, R. C., 2004. A particle swarm optimizer with passive congregation. BioSystems 78, 135–147.

Hendtlass, T., June 2003. Preserving diversity in particle swarm optimisation. In: Chung, P. W. H., Hinde, C. J., Ali, M. (Eds.), Proceedings of the 16th International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, Loughborough, UK, "Developments in Applied Artificial Intelligence" LNAI 2718. Springer-Verlag, Berlin Heidelberg, pp. 31–40.

Ho, S. L., Yang, S., Ni, G., Lo, E. W. C., Wong, H. C., May 2005. A particle swarm optimization-based method for multiobjective design optimizations. IEEE Transactions on Magnetics 41 (5), 1756–1759.

Ho, S. L., Yang, S., Ni, G., Wong, H. C., April 2006. A particle swarm optimization method with enhanced global search ability for design optimizations of electromagnetic devices. IEEE Transactions on Magnetics 42 (4), 1107–1110.

Holland, J. H., 1962. Outline for logical theory of adaptive systems. Journal of ACM 3, 297–314.

Hong, B., Strawderman, R. L., Swaney, D. P., Weinstein, D. A., 2005. Bayesian estimation of input parameters of a nitrogen cycle model applied to a forested reference watershed, Hubbard Brook Watershed Six. Water Resources Research 41, W03007.

Hsu, M.-H., Fu, J.-C., Liu, W.-C., 2006. Dynamic routing model with real-time roughness updating for flood forecasting. Journal of Hydraulic Engineering 132, 605–619.

Huisman, J. A., Breuer, L., Eckhardt, K., Frede, H.-G., 2003. Spatial consistency of automatically calibrated SWAT simulations in the Dill catchment and three of its subcatchments. In: Srinivasan, R., Jacobs, J., Jensen, R. (Eds.), Proceedings of the 2003 International SWAT Conference. Texas Water Resources Institute Technical Report 266, College Station, Texas.

Insightful Corp., 2001. S-Plus Statistical Software: Release 6.0. MathSoft, Seattle, Washington.

Jacquin, A. P., Shamseldin, A. Y., 2007. Development of a possibilistic method for the evaluation of predictive uncertainty in rainfall-runoff modeling. Water Resources Research 43, W04425.

Jha, M., Gassman, P. W., Secchi, S., Gu, R., Arnold, J., 2004. Effect of watershed subdivision on SWAT flow, sediment, and nutrient predictions. Journal of the American Water Resources Association 40 (3), 811–825.

Kalin, L., Govindaraju, R. S., Hantush, M. M., 2003. Effect of geomorphologic resolution on modeling of runoff hydrograph and sedimentograph over small watersheds. Journal of Hydrology 276, 89–111.

Kavetski, D., Franks, S. W., Kuczera, G., 2003. Confronting input uncertainty in environmental modelling. In: Duan, Q., Gupta, H. V., Sorooshian, S., Rousseau, A. N., Turcotte, R. (Eds.), Calibration of Watershed Models. Vol. 6 of Water Science and Application Series. American Geophysical Union, pp. 49–68.

Kennedy, J., 2000. Stereotyping: Improving particle swarm performance with cluster analysis. In: Proceedings of the Congress on Evolutionary Computation. IEEE Service Center, Piscataway, New Jersey, pp. 1507–1512.

Kennedy, J., Eberhart, R. C., 1995. Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks. Piscataway, New Jersey, pp. 1942–1948.

Kennedy, J., Mendes, R., May 2002. Population structure and particle swarm performance. In: Proceedings of the World Congress on Computational Intelligence. Honolulu, Hawaii, pp. 1671–1676.

Khemka, N., Jacob, C., 2004. Exploratory toolkit for evolutionary and swarm-based optimization. In: Proceedings of the 6th International Mathematica Symposium. Banff, Alberta, Canada.

Khorzad, K., October 2003. Edwards aquifer evaluation: Kinney county, Texas. Journal of the American Water Resources Association 39 (5), 1093–1107.

King, K. W., Arnold, J. G., Bingner, R. L., 1999. Comparison of Green-Ampt and curve number methods on Goodwin Creek watershed using SWAT. Transactions of the ASAE 42 (4), 919–925.

Kirkpatrick, S., Gelatt, Jr., C. D., Vecchi, M. P., 1983. Optimization by simulated annealing. Science 220 (4598), 671–680.

Koh, R., Ferguson, W., Kadourri, A., 2004. Bayesian inference using Monte Carlo Markov Chains for optimization and parameter uncertainty assessment in Soil Water Assessment Tool model. In: Liong, S.-Y., Phoon, K.-K., Babovic, V. (Eds.), Proceedings of the 6th International Conference on Hydroinformatics. Vol. 2. World Scientific Publishing Co. Pte. Ltd., pp. 1761–1766.

Krink, T., Vesterstrøm, J. S., Riget, J., May 2002. Particle swarm optimisation with

spatial particle extension. In: Proceedings of the Congress on Evolutionary Computation. Vol. 2. Honolulu, Hawaii, pp. 1474–1479.

Lawler, A. L., 1964. Hydrology of flood control, Part II: Flood routing. In: Chow, V. T. (Ed.), Handbook of Applied Hydrology. McGraw-Hill Book Company, New York, New York, Ch. 25-II.

Lemon, J., Bolker, B., Oom, S., Klein, E., Rowlingson, B., Wickham, H., Tyagi, A., Eterradossi, O., Grothendieck, G., Toews, M., 2007. plotrix: Various plotting functions. R package version 2.2-7.

Li, X., 2004. Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization. Lecture Notes in Computer Science 3102, 105–116.

Lindgren, R. J., Dutton, A. R., Hovorka, S. D., Worthington, S. R. H., Painter, S., 2004. Conceptualization and simulation of the Edwards aquifer, San Antonio region, Texas. US Geological Survey Scientific Investigations Report 2004-5277.

Liong, S. Y., Muttil, N., September 2001. Shuffled complex evolution coupled with experimental design technique. In: Proceedings of the XXIX International Association of Hydraulic Engineering and Research Congress. Beijing, China.

Locatelli, M., 2003. A note on the Griewank test function. Journal of Global Optimization 25, 169–174.

Lv, Y., Li, S., Chen, S., Jiang, Q., Guo, W., 2006. Particle swarm optimization based on information diffusion and clonal selection. Lecture Notes in Computer Science 4247, 521–528.

Madsen, H., 2003. Parameter estimation in distributed hydrological catchment modelling using automatic calibration with multiple objectives. Advances in Water Resources 26, 205–216.

Makowski, D., Wallach, D., Tremblay, M., 2002. Using a Bayesian approach to parameter estimation; comparison of the GLUE and MCMC methods. Agronomie 22, 191–203.

Mamillapalli, S., 1997. Effect of spatial variability in modeling river basin runoff. Ph.D. dissertation, Purdue University, West Lafayette, Indiana.

Mamillapalli, S., Srinivasan, R., Arnold, J. G., Engel, B. A., January 21–26, 1996. Effect of spatial variability on basin scale modeling. In: Proceedings of the 3rd International Conference/Workshop on Integrating GIS and Environmental Modeling, Santa Fe, New Mexico. National Center for Geographic Information and Analysis, Santa Barbara, California.

Manguerra, H. B., Engel, B. A., October 1998. Hydrologic parameterization of watersheds for runoff prediction using SWAT. Journal of the American Water Resources Association 34 (5), 1149–1162.

Mantovan, P., Todini, E., 2006. Hydrological forecasting uncertainty assessment: Incoherence of the GLUE methodology. Journal of Hydrology 330, 368–381.

McIntyre, N., 2004. Analysis of uncertainty in river water quality modelling. Ph.D. dissertation, University of London.

McIntyre, N., Lee, H., Wheater, H., Young, A., Wagener, T., 2005. Ensemble predictions of runoff in ungauged catchments. Water Resources Research 41, W12434.

Meissner, M., Schmuker, M., Schneider, G., March 2006. Optimized particle swarm optimization (OPSO) and its application to artificial neural network training. BMC Bioinformatics 7 (125).

Monson, C. K., Seppi, K. D., 2005. Bayesian optimization models for particle swarms. In: Proceedings of the Genetic and Evolutionary Computation Conference. ACM Press, New York, New York, pp. 193–200.

Monson, C. K., Seppi, K. D., 2006. Adaptive diversity in PSO. In: Proceedings of the Genetic and Evolutionary Computation Conference. ACM Press, New York, New York, pp. 59–66.

Montanari, A., 2005. Large sample behaviors of the generalized likelihood uncertainty estimation (GLUE) in assessing the uncertainty of rainfall-runoff simulations. Water Resources Research 41, W08406.

Montanari, A., 2007. What do we mean by 'uncertainty'? The need for a consistent wording about uncertainty assessment in hydrology. Hydrological Processes 21, 841–845.

Moriasi, D. N., Arnold, J. G., van Liew, M. W., Bingner, R. L., Harmel, R. D., Veith, T. L., 2007. Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. Transactions of the ASABE 50 (3), 885–900.

Muleta, M. K., Nicklow, J. W., 2005. Sensitivity and uncertainty analysis coupled with automatic calibration for a distributed watershed model. Journal of Hydrology 306, 127–145.

Muttiah, R. S., Wurbs, R. A., 2002. Scale-dependent soil and climate variability effects on watershed water balance of the SWAT model. Journal of Hydrology 256,

264–285.

Muttil, N., Liong, S.-Y., 2004. Are complexes in SCE-UA necessary? In: Liong, S.-Y., Phoon, K.-K., Babovic, V. (Eds.), Proceedings of the 6th International Conference on Hydroinformatics. Vol. 1. World Scientific Publishing Co. Pte. Ltd., pp. 874–881.

Nash, J. E., Sutcliffe, J. V., 1970. River flow forecasting through conceptual models, part I–A discussion of principles. Journal of Hydrology 10, 282–290.

Neitsch, S. L., Arnold, J. G., Kiniry, J. R., Williams, J. R., King, K. W., 2002a. SWAT2000 Theoretical Documentation. Texas Water Resources Institute, College Station, Texas.

Neitsch, S. L., Arnold, J. G., Kiniry, J. R., Williams, J. R., King, K. W., 2002b. SWAT2000 User Manual. Texas Water Resources Institute, College Station, Texas.

Neitsch, S. L., Arnold, J. G., Srinivasan, R., February 2002c. Pesticides fate and transport predicted by the Soil and Water Assessment Tool (SWAT): Atrazine, metolachlor and trifluralin in the Sugar Creek watershed. Technical Report 2002-03, Grassland, Soil and Water Research Laboratory, Agricultural Research Service and Blackland Research Center, Texas Agricultural Experiment Station, Temple, Texas.

NOAA-NCDC, 2006. National Oceanic & Atmospheric Administration-National Climatic Data Center. Surface Data, Daily US. http://www.ncdc.noaa.gov, accessed in November, 2006.

Olivera, F., Valenzuela, M., Srinivasan, R., Choi, J., Cho, H., Koka, S., Agrawal, A., 2006. ArcGIS-SWAT: A geodata model and GIS interface for SWAT. Journal of the American Water Resources Association 42 (2), 295–309.

Owen, A. B., 1998. Scrambling Sobol' and Niederreiter-Xing points. Journal of Complexity 14, 466–489.

Parsopoulos, K. E., Plagianakos, V. P., Magoulas, G. D., Vrahatis, M. N., 2001. Stretching technique for obtaining global minimizers through particle swarm optimization. In: Proceedings of the Particle Swarm Optimization Workshop. Indianapolis, pp. 22–29.

Parsopoulos, K. E., Vrabatis, M. N., 2001. Modification of the particle swarm optimizer for locating all the global minima. In: Kurkova, V., Steele, N. C., Neruda, R., Karny, M. (Eds.), Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms. Prague, Czech Republic, pp. 324–327.

Perrin, C., Michel, C., Andréassian, V., 2001. Does a large number of parameters enhance model performance? Comparative assessment of common catchment model structures on 429 catchments. Journal of Hydrology 242, 275–301.

R Development Core Team, 2006. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0. http://www.r-project.org, accessed in January, 2007.

Riget, J., Vesterstrøm, J. S., 2002. A diversity-guided particle swarm optimizer–the ARPSO. Technical Report 2002-02, Department of Computer Science, Aarhus Universitet, Bgn. 540, Ny Munkegade DK-8000 Aarhus C, Denmark.

Rosenthal, W. D., Hoffman, D. W., 1999. Hydrologic modeling/GIS as an aid in locating monitoring sites. Transactions of the ASAE 42 (6), 1591–1598.

Schulz, K., Beven, K., Huwe, B., 1999. Equifinality and the problem of robust calibration in nitrogen budget simulations. Soil Science Society of America Journal 3,

1934–1941.

Shi, Y., Eberhart, R. C., 1998. Parameter selection in particle swarm optimization. In: Proceedings of the 7th International Conference on Evolutionary Programming VII. Springer-Verlag, London, United Kingdom, pp. 591–600.

Sobol', I. M., 1967. The distribution of points in a cube and the approximate evaluation of integrals. USSR Computational Mathematics and Mathematical Physics 7 (4), 86–112.

Socha, K., Dorigo, M., April 2006. Ant colony optimization for continuous domains. Technical Report TR/IRIDIA/2006-037.002, IRIDIA, Université Libre de Bruxelles, CP 194/6, Ave. Franklin D. Roosevelt 50, 1050 Brussels, Belgium.

Soil Conservation Service, 1972. National Engineering Handbook, Section 4. U.S. Department of Agriculture, Washington, District of Columbia.

Sotiropoulos, D. G., Plagianakos, V. P., Vrahatis, M. N., 2002. An evolutionary algorithm for minimizing multimodal functions. In: Lipitakis, E. A. (Ed.), Proceedings of the 5th Hellenic-European Conference on Computer Mathematics and its Applications. LEA Press, pp. 496–500.

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

Stephenson, G. R., Freeze, R. A., 1974. Mathematical simulation of subsurface flow contributions to snowmelt runoff, Reynolds Creek, Idaho. Water Resources Research 10 (2), 284–298.

Suganthan, P. N., July 1999. Particle swarm optimizer with neighborhood operator. In: Proceedings of the Congress on Evolutionary Computation. Washington, District of Columbia, pp. 1958–1961.

Thiemann, M., Trosset, M., Gupta, H., Sorooshian, S., October 2001. Bayesian recursive parameter estimation for hydrologic models. Water Resources Research 37 (10), 2521–2535.

Todini, E., Mantovan, P., 2007. Comment on: 'On undermining the science?' by Keith Beven. Hydrological Processes 21, 1633–1638.

Trabia, M. B., November 2004. A hybrid fuzzy simplex genetic algorithm. Journal of Mechanical Design 126, 969–974.

Tripathi, M. P., Panda, R. K., Raghuwanshi, N. S., 2003. Identification and prioritization of critical sub-watersheds for soil conservation management using the SWAT model. Biosystems Engineering 85 (3), 365–379.

Tripathi, M. P., Raghuwanshi, N. S., Rao, G. P., 2006. Effect of watershed subdivision on simulation of water balance components. Hydrological Processes 20, 1137–1156.

USDA-NRCS, 2006. U.S. Department of Agriculture-Natural Resources Conservation Service. State Soil Geographic Database. http://www.ncgc.nrcs.usda.gov/products/datasets/statsgo, accessed in November, 2006.

USEPA, 2007. U.S. Environmental Protection Agency. Reach File 1 (RF1) for the Conterminous United States in BASINS. http://www.epa.gov/waterscience/BASINS/metadata/rf1.htm, accessed in January, 2007.

USGS, 2006a. U.S. Geological Survey. National Elevation Dataset (NED). http://ned.usgs.gov, accessed in November, 2006.

USGS, 2006b. U.S. Geological Survey. National Land Cover Dataset 1992 (NLCD 1992). http://landcover.usgs.gov/natlandcover.php, accessed in November, 2006.

USGS, 2006c. U.S. Geological Survey. Surface-Water Daily Data for the Nation. http://waterdata.usgs.gov/nwis, accessed in November, 2006.

USGS, 2007. U.S. Geological Survey. Daily Values of Suspended Sediment and Ancillary Data. http://co.water.usgs.gov/sediment/seddatabase.cfm, accessed in January, 2007.

van den Bergh, F., 2002. An analysis of particle swarm optimizers. Ph.D. dissertation, Department of Computer Science, University of Pretoria, Pretoria, South Africa, cited in Brits et al. (2007).

van den Bergh, F., Engelbrecht, A. P., 2002. A new locally convergent particle swarm optimiser. In: Proceedings of the IEEE Conference on Systems, Man and Cybernetics. pp. 96–101.

van Griensven, A., Bauwens, W., 2001. Integral water quality modelling of catchments. Water Science & Technology 43 (7), 321–328.

Wang, X., Gao, X. Z., Ovaska, S. J., 2007. A hybrid optimization algorithm based on ant colony and immune principles. International Journal of Computer Science & Applications 4 (3), 30–44.

Weber, A., Fohrer, N., Möller, D., 2001. Long-term land use changes in a mesoscale watershed due to socio-economics factor—effects on landscape structures and functions. Ecological Modelling 140, 125–140.

Wei, L., Zhao, M., 2005. A niche hybrid genetic algorithm for global optimization of continuous multimodal functions. Applied Mathematics and Computation 160,

649–661.

Williams, J. R., 1969. Flood routing with variable travel time or variable storage coefficients. Transactions of the ASAE 12 (1), 100–103.

Williams, J. R., 1995. The EPIC model. In: Singh, V. P. (Ed.), Computer Models of Watershed Hydrology. Water Resources Publications, Littleton, Colorado, Ch. 25, pp. 909–1000.

Würtz, D., 2004. Rmetrics: An Environment for Teaching Financial Engineering and Computational Finance with R. Rmetrics, ITP, ETH Zürich, Zürich, Switzerland, http://www.rmetrics.org, accessed in January, 2007.

Xie, X.-F., Zhang, W.-J., Yang, Z.-L., May 2002. A dissipative particle swarm optimization. In: Proceedings of the Congress on Evolutionary Computation. Honolulu, Hawaii, pp. 1456–1461.

Xiong, L., O'Connor, K. M., 2000. Analysis of the response surface of the objective function by the optimum parameter curve: How good can the optimum parameter values be? Journal of Hydrology 234, 187–207.

Yang, X., Yang, Z., Lu, G.-H., Li, J., 2005. A gray-encoded, hybrid-accelerated, genetic algorithm for global optimizations in dynamical systems. Communications in Nonlinear Science and Numerical Simulation 10, 355–363.

Yapo, P. O., Gupta, H. V., Sorooshian, S., 1998. Multi-objective global optimization for hydrologic models. Journal of Hydrology 204, 83–97.

Zechman, E. M., Ranjithan, R. S., 2003. Are the "best" solutions to a real optimization problem always found in the noninferior set? Evolutionary algorithm for

generating alternatives (EAGA). Lecture Notes in Computer Science 2724, 1622–1623.

Zechman, E. M., Ranjithan, R. S., March 2007. Generating alternatives using evolutionary algorithms for water resources and environmental management problems. Journal of Water Resources Planning and Management 133 (2), 156–165.

Zhang, Q., Sun, J., Tsang, E., Ford, J., 2004. Hybrid estimation of distribution algorithm for global optimization. Engineering Computations 21 (1), 91–107.

Zheng, Y., Keller, A. A., 2007. Uncertainty assessment in watershed-scale water quality modeling and management: 1. Framework and application of generalized likelihood uncertainty estimation (GLUE) approach. Water Resources Research 43, W08407.

APPENDIX A

DIRECTORY STRUCTURE OF THE OPTICAL DISC

The following list describes the directory structure of the optical disc accompanying with this dissertation:

- /libsce contains the C implementation of the Shuffled Complex Evolution (SCE-UA) algorithm used for the calibration performed in Chapter II (see Appendix B for more details);
- /spatial_variability contains the calibrated and validated models used for the analysis performed in Chapter II (see Appendix C for more details);
- /ispso contains the R implementation of ISPSO used in Chapter III and a sample script (see Appendix D for more details);
- /griewank contains the R code used to count the number of minima of the Griewank function in Chapter IV (see Appendix E for more details);
- /prediction_uncertainty contains the sample models used for the uncertainty analysis performed in Chapter V (see Appendix F for more details).

APPENDIX B

LIBSCE

B.1. Introduction

The libsce library is a C program developed by Huidae Cho for his doctoral research and provides a framework for the Shuffled Complex Evolution algorithm developed at the University of Arizona (SCE-UA) (Duan et al., 1993). Using this library, researchers can develop their own calibration tool for a specific model. The original SCE-UA program was written in Fortran 90. The main goals of libsce are two-fold: (1) to develop a completely independent library so that the user does not need to modify the optimization source code in the library or the model to be linked with the library and (2) to allow multiple parameter values for each parameter in the SCE-UA algorithm in order to calibrate distributed hydrologic models.

Section B briefly describes how libsce works, Section B explains how to install the library, and Section B introduces two sample calibration tools developed using libsce.

B.2. How it works

The model to be linked with libsce should be called explicitly by the calibration tool, and its input files also need to be rewritten. Figure 28 shows the protocol used for communication among three parts: the calibration tool, libsce, and the model. The calibration tool behaves as the interface between libsce and the user. It provides a file to configure various control parameters for optimization and model information.

It also receives requests from the library, runs the model, and performs the post-processing of the model output files such as extracting required data, calculating errors, and saving them to files for reference.

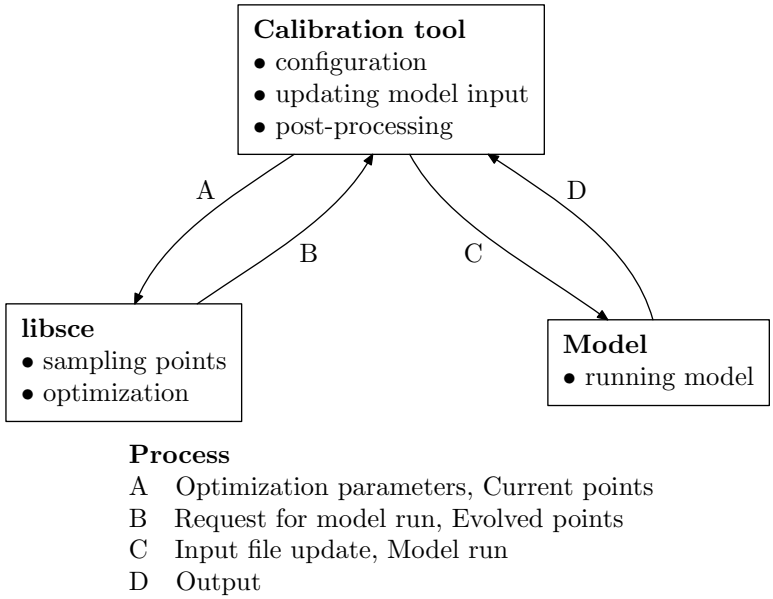


Figure 28: Protocol among calibration tool, libsce, and model.

Given optimization specifications and model information, the library takes samples and evolves them using the SCE-UA algorithm while sending requests for model runs to the calibration tool. This procedure is the core of the library, and the user does not need to modify its source code at all by its design. The model part is simply the model itself. Whenever it is executed, input files are updated by the calibration tool, and its output is analyzed by user-specified post-processors.

As shown in Figure 29, libsce has three data structures (struct in the C language). The sce structure has all the parameters required by the SCE-UA algorithm. These parameters include the number of consecutive offsprings generated by each sub-complex, the number of points in a complex, etc. The lower and upper bounds of the model parameters and the variables of the parameter function are stored in the

parameter structure. The most complicated part is the model structure because it is the actual interface between the model and libsce. It has pointers to the user-defined functions that read and update the model input files, run the model, and calculate model errors. Provided the function pointers, the sce_method routine is able to access the model through these pointers. For full description of these structures, see Tables 18, 19, and 20. Note that the actual names of these structures have an underbar "_" prefix to allow the user to define variables such as sce, param, and model instead of inventing other names.

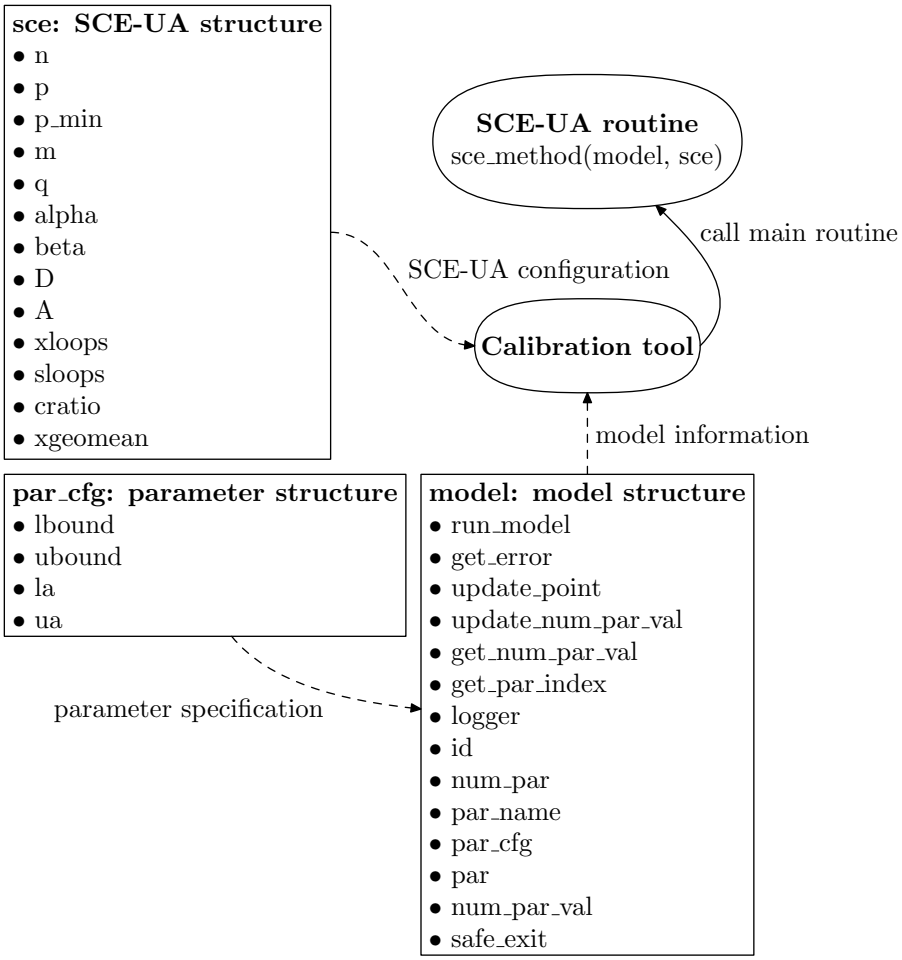


Figure 29: libsce data structures.

The calibration tool is a separate program from the model, which means that it should know about the model parameters before running the model. To provide model information to the calibration tool, four members of the model structure are used: num_par, par_name, par_cfg, par, and num_par_val. The parameter specifications are not that simple for distributed hydrologic models. For spatially distributed hydrologic models, there are usually more than one value per parameter because each spatial unit has its own value. It means that one parameter in the model does not necessarily have one number in the optimization algorithm because only one value cannot represent all the units that are spatially distributed. For example, 10 subwatersheds have 10 different curve numbers even if they all represent one parameter called the curve number. To support models like this, the concept of the parameter dimension is introduced. In the previous example, the curve number has a dimension of 10 because there are 10 subwatersheds with different curve numbers. However, even if the model is a distributed model, it can have parameters having only one value such as the watershed area. That is, different parameters can have different dimensions. There are three functions related to this feature: update_num_par_val, get_num_par_val, and get_par_index. One point in the library stores a set of parameter values for one model run. One point has $\sum_{i=1}^{n} dim_i$ parameter values where n is the number of parameters, and dim_i is the dimension of the parameter i. To store all these values, the array D is used, and these values are partitioned into p complexes and saved in the array A.

In general, the calibration tool requires a large number of model runs regardless of its algorithm, which means that optimization takes much time to find a good solution. However, in some cases, the user may want to stop the process even if it is still running when he or she is already satisfied with current results or computing resource is limited. If the process quits without caution, precious information can be lost because the last input files are not always the best estimates so far. Further, the

Table 18: SCE-UA data structure: struct _sce.

Member	Description
n	Number of parameters
p	Number of complexes
p_min	Minimum number of complexes required in the population
m	Number of points in a complex
q	Number of points in a sub-complex
alpha	Number of consecutive offspring generated by each sub-complex
beta	Number of evolution steps taken by each complex
D	All points are stored in this array
A	D is divided into p complexes and saved in this array
xloops	Maximum number of loops
sloops	Number of shuffling loops in which the change in criterion value is expected
cratio	Ratio by which the criterion value must change in the given number of shuffling loops (xloops)
xgeomean	Maximum geometric mean of all parameter spaces to determine if the population has converged sufficiently

calibration tool cannot save all input files created in every generation of optimization due to the limitation of data storage. For this reason, the user cannot expect that the best input files are stored somewhere while the program is running. To get around this problem, the safe_exit function is introduced that traps all termination requests and saves the best solution to files.

Table 19: Model data structure: struct _model.

Member	Description
run_model	Function pointer to run the model
get_error	Function pointer to calculate errors
update_point	Function pointer to compute new point values and update input files
update_num_par_val	Function pointer to update the array of the number of parameter values
get_num_par_val	Function pointer to get the dimension of a parameter
get_par_index	Function pointer to return which parameter is used in the given index of a point
logger	Function pointer to write to a file process information
id	Identification number used to print what is being optimized
num_par	Number of parameters
par_name	Array of parameter names
par_cfg	Pointer to _par_cfg struct
par	Array of parameters to be optimized
num_par_val	Array of the number of parameter values
safe_exit	Function pointer to save the best estimates before sudden user interruption

The logger function is also provided for the purpose of debugging. It would be helpful for the user to debug the calibration tool or even to correct optimization specifications while running the program. It is impossible that the user sits in front of the computer all the time and waits for long optimization processes to finish. The calibration tool will generate a log file to save information about each step of the

progress so that the user can easily diagnose what is going on when some things go wrong.

Table 20: Parameter data structure: struct _par_cfg.

Member	Description
lbound	Lower bound of the parameter
ubound	Upper bound of the parameter
la	Lower bound of the parameter function variable
ua	Upper bound of the parameter function variable

There are eight functions that the user has to provide (run_model, get_error, update_point, update_num_par_val, get_num_par_val, get_par_index, logger, and safe_exit) and six custom functions (main, read_config, read_files, update_par, get_value, and calibrate). The flow of function calls is presented in Figure 30. The get_par_index function is reserved for future use, and the logger and safe_exit functions are executed by the ANSI C signal handlers when requested by the user.

B.3. Installation

Libsce is written in the C language, so the reader is supposed to be familiar with C programming. The library was tested with the GNU C compiler and the GNU make utility, so other compilers might not compile the sources. The library is designed mainly for UNIX-like operating systems to facilitate fast optimization on supercomputers. However, it is also built on Cygwin, a UNIX emulator on MS-Windows.

The steps to install the library are as follows: (1) uncompress /libsce/libsce.tgz: `gzip -dc libsce.tgz | tar -xvf -`, (2) compile the library: `cd libsce && make`, and (3) copy libsce.a and sce.h to an appropriate directory.

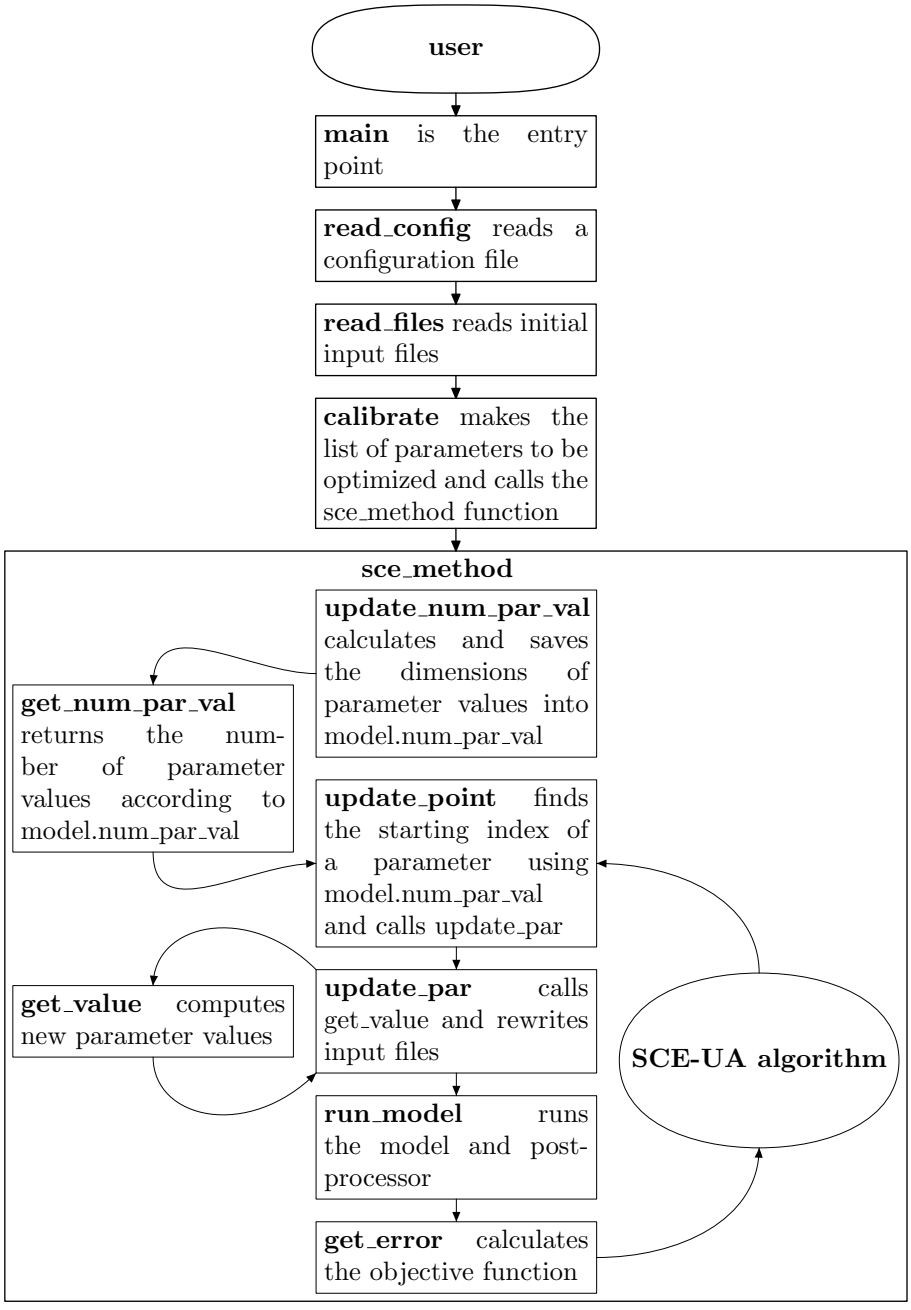


Figure 30: Flowchart for function calls in calibration.

B.4. Sample calibration tools

Two sample calibration tools are included: sce4tmod (an SCE-UA calibration tool for a TOPMODEL version, TMOD9502) and sce4swat (an SCE-UA calibration tool for SWAT 2000). The compressed files for the two calibration tools are sce4tmod.tgz and sce4swat.tgz, respectively, in the /libsce directory.

The steps to build the executable file for either sce4tmod or sce4swat are as follows: (1) compile and install the libsce library, (2) uncompress /libsce/sce4swat.tgz: `gzip -dc sce4swat.tgz | tar -xvf -`, and (3) compile the calibration tool: `cd sce4swat && make`.

APPENDIX C

CALIBRATED AND VALIDATED MODELS USED FOR THE ANALYSIS IN CHAPTER II

The calibrations in Chapter II were performed using an automatic calibration program called sce4swat introduced in Appendix B. The contents of the /spatial_variability directory are as follows:

- models contains the SWAT models used for the analysis in Chapter II;
- observed_flows.tgz contains the observed daily streamflow data for the six USGS gauges shown in Figures 1 and 2 for two periods (i.e., calibration and validation);
- simulated_flows.tgz contains the simulated daily streamflows at the watershed outlets and the simulated daily runoff flows from the land surfaces.

The naming convention for the compressed files in the models directory is *MUPLS*.tgz where *M* is "u" (uncalibrated), "c" (calibrated), or "v" (validated); *U* is 08070200 (the East Fork of the San Jacinto River watershed), 08155240 (the Barton Creek watershed), and 08159000 (the Onion Creek watershed); *P* is "o" (multiple rain gauges) or "s" (single rain gauges); *L* is "o" (original land use), "d" (single land use), or "r" (random land use); *S* is "o" (original soil type), "d" (single soil type), or "r" (random soil type); and .tgz is the file extension for gzipped tar files. For example, the file v08159000sdr.tgz contains the model of the Onion Creek watershed with a single rain gauge, a single land use, and a random land use. To uncompress gzipped tar files, execute the following command: `gzip -dc v08159000sdr.tgz | tar -xvf -`.

The naming convention for the text files in the observed_flows.tgz file is *MU*.txt where *M* and *U* follow the same rules as above.

The simulated_flows.tgz file contains three types of simulated flow data: the simulated streamflows obtained from the models in the models directory, the runoff flows obtained from the same directory, and the simulated streamflows obtained by varying the Manning's channel roughness coefficient (CH_N2 in the SWAT documentation). The simulated streamflows extracted from the models in the models directory follow the same naming convention as the model itself (i.e., *MUPLS*.txt). The simulated runoff flows append the "wyld_" prefix to their corresponding model names. The simulated streamflows calculated by varying the Manning's channel roughness coefficient append the ".ch_n2={*CH_N2*}" suffix to the model names where {*CH_N2*} is a value of the Manning's channel roughness coefficient; for example, the file c08159000sro.ch_n2=0.010.txt contains the simulated streamflows of the c08159000-sro model with a CH_N2 value of 0.010.

APPENDIX D

ISPSO.R

D.1. Introduction

ISPSO.R is an R implementation of the Isolated Speciation-based Particle Swarm Optimization (ISPSO) introduced in Chapter III. Since the R language (R Development Core Team, 2006) is an open-source counterpart of the S-PLUS statistics package (Insightful Corp., 2001), pre-compiled binary distributions for Linux, MacOS X, and MS-Windows are freely available from its website http://www.r-project.org.

Section D explains how to install R and ISPSO.R, and Section D documents how to run the program.

D.2. Installation

Installation steps for R are as follows: (1) download the base R system from the Comprehensive R Archive Network (CRAN) listed under the download section of R's website and (2) uncompress the downloaded file if necessary, and run the executable file.

ISPSO.R requires several R packages: a package collection called Rmetrics (Würtz, 2004) and plotrix (Lemon et al., 2007). Because R automatically detects interdependency between packages and installs required packages, several packages in the Rmetrics package collection can be installed by installing a package called fOptions, which ISPSO.R directly calls. The steps to install fOptions and plotrix are as follows: (1) execute R and (2) use the install.packages function to install the two packages: `install.packages(c("fOptions", "plotrix"))`.

Sample files are provided with ISPSO.R: uncompress /ispso/ispso.tgz: `gzip -dc ispso.tgz | tar -xvf -`. There are three files in the compressed file: ispso.R defines the pso function that implements the ISPSO algorithm, funcs.R provides test functions, and test.R is a sample file that executes the pso function for test.

D.3. How to run

The pso function takes a list as its input, the members of which include debugging parameters and the ISPSO control parameters. The debugging parameters are explained in detail in the test.R file and, in this section, only the ISPSO control parameters are shown in Table 21. This function returns a list as its output, the members of which are shown in Table 22. The pso function is called as shown in Figure 31. After creating an R script file, execute the file from R: `source("test.R")`.

Table 21: ISPSO control parameters.

Parameter	Data type	Description
mycontrib	Constant array	ISPSO
pso	Constant string	"spso"
vupdate	Constant string	"constriction"
c1	Real	Constriction coefficient ψ_1 in Eq. (3.3)
c2	Real	Constriction coefficient ψ_2 in Eq. (3.3)
w	Real	Constriction coefficient χ in Eq. (3.3)
S	Integer	Swarm size $\|S\|$
vmax	Array of real numbers	Maximum velocities $\vec{x}_{\max}$
vmax0	Real	Maximum initial velocity $\|\vec{v}_{\max,0}\|$
maxiter	Integer	Number of the maximum iterations
xeps	Real	Nesting criterion ϵ_x
feps	Real	Nesting criterion ϵ_f
age	Integer	Nesting criterion, particle's age a
rspecies	Real	Species radius r_{species}
rprey	Real	Prey radius r_{prey}
rnest	Real	Nest radius r_{nest}
f	Function	Problem to solve
D	Integer	Problem dimension
xmin	Array of real numbers	Lower bounds of the problem domain
xmax	Array of real numbers	Upper bounds of the problem domain

```
# Include ispso.R.
source("ispso.R")
# Define test functions.
source("funcs.R")

s <- list()

# Stop after all the solutions are found.
s$.stop_after_solutions <- -1

########################################
# Do not touch the following variables.
s$mycontrib <- ISPSO
s$pso <- "spso"
s$vupdate <- "constriction"
########################################

# Constriction coefficients
s$c1 <- 2.05
s$c2 <- 2.05
s$w <- 2/abs(2-s$c1-s$c2-sqrt((s$c1+s$c2)^2-4*(s$c1+s$c2)))

# Problem to solve
s$f <- f5
s$D <- 2

# Domain space
s$xmin <- rep(-6, s$D)
s$xmax <- rep(6, s$D)

# Velocities
s$vmax <- (s$xmax-s$xmin)*0.1
s$vmax0 <- sqrt(sum((s$xmax-s$xmin)^2))*0.001

# Radii
s$rspecies <- sqrt(sum((s$xmax-s$xmin)^2))*0.1
s$rprey <- sqrt(sum((s$xmax-s$xmin)^2))*0.0001
s$rnest <- sqrt(sum((s$xmax-s$xmin)^2))*0.01

# Nesting criteria
s$xeps <- 0.001
s$feps <- 0.0001
s$age <- 10

# Swarm size
s$S <- 10 + floor(2*sqrt(s$D))

# Number of maximum iterations
s$maxiter <- 2000

# Call the pso function.
ret <- pso(s)
```

Figure 31: How to call the pso function.

Table 22: ISPSO return variables.

Parameter	Data type	Description
iter	Integer	Number of iterations
evals	Integer	Number of function evaluations
nest	Matrix	Solutions found. Columns x1, x2, $\cdots$, xD: the coordinates of the solution; column f: the objective function value; column v: the particle's velocity; column age: the particle's age; column evals: the number of function evaluations required to find this solution
pop	Matrix	All parameter samples. Columns x1, x2, $\cdots$, xD: the coordinates of the sample; column f: the objective function value; column v: the particle's velocity; column age: the particle's age

APPENDIX E

R CODE USED TO COUNT THE NUMBER OF MINIMA OF THE GRIEWANK FUNCTION

E.1. Introduction

This appendix introduces the set of R script files used to count the number of minima of the Griewank function as explained in Chapter IV. The following section describes how to install and run these R scripts.

E.2. How to run

These R script files require the R base system introduced in Appendix D. After the installation of R, uncompress /griewank/griewank.tgz: `gzip -dc griewank.tgz | tar -xvf -`.

There are four R script files: (1) funcs.R defines the Griewank function and the function that counts the number of minima given a domain space, (2) find_xmax.R tries to find a hyperrectangle that the method introduced in Chapter IV can be applied to, (3) check_xmax.R plots the cosine and sine curves of the boundaries of the hyperrectangle, and (4) doit.R calls find_xmax.R and check_xmax.R to find appropriate hyperrectangles for up to 50 dimensional problems. There is no need to modify the source code of any script file, and running doit.R from R is enough: `source("doit.R")`.

APPENDIX F

SAMPLE MODELS USED FOR THE ANALYSIS IN CHAPTER V

The contents of the /prediction_uncertainty directory are as follows:

- observed.tgz contains the observed daily streamflow and sediment discharge data for three periods (i.e., calibration, rejection, and verification);
- c08019500.11.tgz contains the coordinates, the objective function values, and the likelihood measures of 46,000 samples used for the ISPSO-GLUE approach in Chapter V;
- simulated.11c.tgz contains the simulated daily streamflow and sediment discharge data of the 46,000 samples above;
- r08019500.11.tgz contains the coordinates, the objective function values, and the likelihood measures of 9,622 behavioral models after calibration in the rejection period for the ISPSO-GLUE approach in Chapter V;
- simulated.11r.tgz contains the simulated daily streamflow and sediment discharge data of the 9,622 samples above;
- v08019500.11.tgz contains the coordinates, the objective function values, and the likelihood measures of 499 behavioral models after rejection in the verification period for the ISPSO-GLUE approach in Chapter V;
- simulated.11v.tgz contains the simulated daily streamflow and sediment discharge data of the 499 samples above;

- c08019500.glue.tgz contains the coordinates, the objective function values, and the likelihood measures of 46,000 samples used for the GLUE approach in Chapter V;
- simulated.gluec.tgz contains the simulated daily streamflow and sediment discharge data of the 46,000 samples above;
- r08019500.glue.tgz contains the coordinates, the objective function values, and the likelihood measures of 1,115 behavioral models after random sampling for the GLUE approach in Chapter V;
- simulated.gluer.tgz contains the simulated daily streamflow and sediment discharge data of the 1,115 samples above;
- v08019500.glue.tgz contains the coordinates, the objective function values, and the likelihood measures of 105 behavioral models after rejection for the GLUE approach in Chapter V;
- simulated.gluev.tgz contains the simulated daily streamflow and sediment discharge data of the 105 samples above.

The naming convention for the files in observed.tgz is *V*_*M*08019500.txt where *V* is "flow" (daily streamflow) or "sed" (daily sediment discharge); and *M* is "c" (the calibration period), "r" (the rejection period), or "v" (the verification period). The text files in simulated.*.tgz have the same rule except that the _*N* suffix is appended where *N* is a 5-digit sample number with left-padding 0's.

The *M*08019500.*.tgz files, where *M* is the same as above, contains SWAT input files where the current 19 parameter values are obtained from the last sample of each simulation period, alpha.txt (the 17 α values of samples), and f.txt ($1-\mathrm{NS}_Q$, $1-\mathrm{NS}_S$, GOC, and $L(\theta|\xi, y)$ of the samples as defined in Chapter V).

VITA

Huidae Cho earned his Bachelor of Science degree in civil engineering from Kyungpook National University, Daegu, Korea, in March 1999 and received his Master of Science degree with specialty in computational hydrology from the same university in February 2001. In 2000, from February to October, he provided lectures on C programming, CGI/PHP programming, and UNIX server at the Computer Center of Kyungpook National University. In June 2000, he has joined the GRASS GIS Development Team by contributing a significant amount of source code. From December 2000 to July 2003, he served as the software development team leader in a software company that provides knowledge-based management systems and electronic document management solutions. In August 2003, he entered the Graduate School of Texas A&M University, College Station, Texas, after winning a scholarship for study abroad graduate students from the Korea Ministry of Science & Technology. While in graduate school, he worked as a graduate research assistant. In August 2007, he was awarded the department head fellowship. He earned his Ph.D. degree from Texas A&M University in August 2008. His research interests include optimization theory, uncertainty analysis, and computational hydrology. He is a registered engineer in Korea and is a member of the American Water Resources Association, of the American Geophysical Union, of the Open Source Geospatial Foundation, and of the GRASS GIS Development Team. His permanent address is WERC 205W, Department of Civil Engineering, Texas A&M University, 3136 TAMU, College Station, TX 77843-3136, and his email address is hcho.eng@gmail.com.

This document was typeset in LaTeX by Huidae Cho.