Dynamically dimensioned search algorithm for computationally efficient watershed model calibration

Bryan A. Tolson[1] and Christine A. Shoemaker[2]

Received 10 November 2005; revised 25 May 2006; accepted 31 August 2006; published 17 January 2007.

[1] A new global optimization algorithm, dynamically dimensioned search (DDS), is introduced for automatic calibration of watershed simulation models. DDS is designed for calibration problems with many parameters, requires no algorithm parameter tuning, and automatically scales the search to find good solutions within the maximum number of user-specified function (or model) evaluations. As a result, DDS is ideally suited for computationally expensive optimization problems such as distributed watershed model calibration. DDS performance is compared to the shuffled complex evolution (SCE) algorithm for multiple optimization test functions as well as real and synthetic SWAT2000 model automatic calibration formulations. Algorithms are compared for optimization problems ranging from 6 to 30 dimensions, and each problem is solved in 1000 to 10,000 total function evaluations per optimization trial. Results are presented so that future modelers can assess algorithm performance at a computational scale relevant to their modeling case study. In all four of the computationally expensive real SWAT2000 calibration formulations considered here (14, 14, 26, and 30 calibration parameters), results show DDS to be more efficient and effective than SCE. In two cases, DDS requires only 15–20% of the number of model evaluations used by SCE in order to find equally good values of the objective function. Overall, the results also show that DDS rapidly converges to good calibration solutions and easily avoids poor local optima. The simplicity of the DDS algorithm allows for easy recoding and subsequent adoption into any watershed modeling application framework.

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

1. Introduction

[2] Almost all watershed simulation models contain effective physical and/or conceptual model parameters that are either difficult or impossible to directly measure. Applications of these models therefore require that model parameters are adjusted so that model predictions closely replicate the observed environmental system response data. The process of model parameter conditioning to historical system response data is called calibration. The traditional approach to model calibration has been to calibrate the model manually by trial and error. While such a manual calibration is useful as a learning exercise for modelers, it can be extremely labor intensive and difficult to implement for complex model calibration situations where models are calibrated to long time series of measured system response data with different constituents at multiple locations.

[3] Watershed modelers have long since recognized that optimization algorithms could be used to automate the calibration process. Automatic calibration is defined here as an optimization algorithm based search for a set of watershed model parameter values that minimize the model prediction errors relative to available measured data for the system being modeled. This study will focus on the automatic calibration of watershed simulation models. The results of this study however are also relevant to all other environmental simulation models requiring calibration. *Gupta et al.* [1998] and *Singh and Woolhiser* [2002] note that the automatic calibration methodology has a number of important parts including: (1) the selection of appropriate calibration data, (2) the definition of the objective function that measures the error between model predictions and the calibration data, and (3) the optimization algorithm used to optimize the selected objective function. This study is focused on investigating optimization algorithms for automatic calibration and in particular will introduce a new and efficient algorithm called the dynamically dimensioned search (DDS).

[4] Early automatic calibration studies utilized local optimization techniques that find locally optimal solutions close to the initial solution [*Ibbitt*, 1970; *Nash and Sutcliffe*, 1970; *Sorooshian and Gupta*, 1983]. Examples include derivative-based (e.g., quasi-Newton) algorithms or derivative free algorithms like the Nelder-Mead Simplex method [*Nelder and Mead*, 1965]. The problem with these methods is that they may find only a local optimum and never get close to the global optimum. Given the inherent complexity of watershed models, recent studies have utilized more

[1]Department of Civil and Environmental Engineering, University of Waterloo, Waterloo, Ontario, Canada.
[2]School of Civil and Environmental Engineering, Cornell University, Ithaca, New York, USA.

0043-1397/07/2005WR004723

advanced global search methods. *Duan* [2003] provides a good review of optimization algorithms for watershed model calibration and his list of global optimization algorithms applied to watershed model calibration includes adaptive random sampling [*Masri et al.*, 1980], controlled random search [*Price*, 1978], the multistart Simplex, genetic algorithm [*Wang*, 1991], simulated annealing [*Thyer et al.*, 1999], and the shuffled complex evolution (SCE) algorithm [*Duan et al.*, 1993, 1992]. SCE is the dominant optimization algorithm in the watershed model automatic calibration literature over the past 10 years given that more than 300 different publications reference the original set of SCE publications [*Duan et al.*, 1993, 1992, 1994]. Therefore our new DDS algorithm is tested extensively against SCE.

[5] The introduction of SCE for automatic calibration of watershed models was a great advancement that has enabled a substantial number of modelers to solve difficult calibration problems. A review of the algorithm performance comparisons in the watershed modeling literature shows that the SCE algorithm was judged to outperform the other global optimization algorithms in the previous paragraph in at least one study (and often multiple studies). However, most of these SCE comparisons involved computationally efficient lumped parameter conceptual watershed models with simulation times often on the order of a few seconds or less. As a result, most previous SCE comparisons utilize very large numbers of total allowable model evaluations per optimization trial. For example, in three studies calibrating 11–13 model parameters, SCE results were generated using 11,000 to 23,000 model evaluations [*Duan et al.*, 1994; *Gan and Biftu*, 1996; *Sorooshian et al.*, 1993]. In more complex model calibration examples, *Tanakamaru and Burges* [1996] used 39,000 to 49,000 model evaluations for SCE in a 16 parameter problem, while *Franchini et al.* [1998] use 250,000 model evaluations in a 37 parameter problem. Consider that the Soil and Water Assessment Tool version 2000 (SWAT2000) distributed watershed model calibration case study utilized here (see section 2.4) requires at least 2 minutes to execute a single, 9-year, daily time step simulation on a Pentium IV 3-GHz processor. Therefore one SCE optimization run in this situation would require about 14 days of computation time for 10,000 SWAT model evaluations and about 4.6 months for 100,000 model evaluations. With such extreme computational burdens in mind, this study is focused on evaluating optimization algorithm performance on rather limited computational budgets (1000 to 10,000 model evaluations).

[6] One approach to address this SCE efficiency issue is to simply run SCE for as long as the case study specific computational constraints allow for (e.g., $\sim$1000 rather than 100,000 simulations). While this approach will produce results, and perhaps even a seemingly reasonable objective function value, SCE was not specifically developed and tested against other algorithms from this perspective. Instead, SCE was developed so that optimal or near-optimal solutions are returned with high reliability upon algorithm convergence (typically more than 10,000 model evaluations). The available SCE comparison literature almost exclusively presents algorithm performance comparisons in terms of effectiveness (solution quality) and computational effort required to find the final best solutions at algorithm termination or convergence but do not assess algorithm effectiveness prior to termination [see, e.g., *Duan et al.*, 1993; *Gan and Biftu*, 1996; *Franchini et al.*, 1998]. This comparison approach is entirely appropriate given that the hydrologic models being calibrated in these case studies were lumped parameter conceptual models with very short simulation times.

[7] When automatic calibration is applied to spatially distributed models, or more generally any model that presents a significant computational burden, the comparison of two optimization algorithms must consider how solution quality changes with varying computational effort. This is because distributed modeling computational timescales can vary by many orders of magnitude depending on what model, spatial discretization level and watershed size is selected for the modeling case study. *Singh and Woolhiser* [2002] report in their review of mathematical modeling of watershed hydrology that many current watershed hydrology models are spatially distributed. In fact, as soon as one considers a limited number of model evaluations perspective, the idea of achieving global optimality becomes unreasonable in most automatic calibration problems. As a result, the methods for comparing algorithm performance in this paper are necessarily different from the methods found in the great majority of previous SCE literature. In addition, we believe that improved automatic calibration optimization algorithms can be developed with such a perspective in mind and introduce the new DDS as one such algorithm focused on identifying good calibration solutions when model evaluations are limited.

[8] The specific goals of this study are (1) to introduce the new DDS algorithm for watershed model calibration and (2) present DDS and SCE comparative algorithm performance results in ways that are meaningful for modelers subject to a wide range of computational limitations. DDS requires essentially no parameter tuning and the search strategy is scaled to the user-specified maximum number of objective function evaluations in order to return good solutions across a range of computational limitations. Numerical results will show that DDS is robust and effective and it outperforms the SCE algorithm for real SWAT2000 watershed simulation model calibration formulations of 14, 26, and 30 parameters limited to 10,000 or fewer total model evaluations. This study limits DDS algorithm comparisons to the SCE algorithm because SCE is so frequently applied to hydrologic or watershed simulation model calibration.

[9] The remainder of the paper is organized as follows. Section 2.1 highlights the benchmark optimization algorithms utilized in this study, and the DDS algorithm is described in detail in section 2.2. The optimization test problems and SWAT2000 automatic calibration case studies are introduced in sections 2.3 and 2.4, respectively. All algorithm comparison results are provided in section 3, while section 4 summarizes and highlights the significance of the results. Conclusions and future research directions are detailed in section 5.

2. Methodology

2.1. Benchmark Optimization Algorithms

[10] The main focus of this study is the introduction of the DDS algorithm (see section 2.2) and the subsequent

performance comparisons with alternative algorithms, including the SCE algorithm. The Matlab© (R13) optimization toolbox implementations of a derivative-based optimization algorithm (Matlab fmincon function) and a Nelder-Mead Simplex algorithm (Matlab fminsearch function) were applied to a subset of model calibration problems in order to confirm that they were in fact difficult multimodal optimization problems. The Matlab implemented derivative-based search (referred to as "Fmincon" in the remainder of this paper) as utilized here approximates derivatives by finite differences and implements sequential quadratic programming to find a local minimum.

[11] The Simplex and Fmincon algorithms are local optimization methods. In order to utilize them for global optimization, they were applied as multistart algorithms. For example, in each Simplex or Fmincon optimization trial with multiple restarts, once an algorithm run converged and stopped, another run was started at a different initial solution that was randomly selected. This was repeated until the maximum function evaluation limit was reached or it was clear that restarting the Simplex or Fmincon algorithm would not significantly improve the current best solutions. Since the Simplex algorithm is an unconstrained optimization algorithm, bound constraints on the decision variables were accounted for by using a penalty function approach that assigns infeasible solutions objective function values that are worse than the worst feasible solution found so far.

[12] In this study, the original SCE algorithm was recoded in Matlab and used to generate almost all SCE results reported here. Tests against the original Fortran SCE version (referred to as SCE-UA for "University of Arizona") on multiple test functions confirmed both SCE implementations were consistent. The original Fortran-coded SCE algorithm is referred to as SCE-UA where appropriate in the remainder of this paper. Besides the random number generation routines, the only other difference between our Matlab SCE and SCE-UA is that our Matlab SCE implementation only stops when the maximum function evaluation limit is reached.

[13] SCE is applied to all test functions and calibration formulations considered in this study. An overview of the algorithm based largely on the summary by *Duan et al.* [1992] is as follows. SCE is a probabilistic population-based evolutionary type of algorithm. The initial population is sampled randomly from the space of feasible solutions. The population is divided into a number of subpopulations called complexes after sorting the population based on objective function value. Each complex is evolved (i.e., improved) using the competitive complex evolution (CCE) algorithm which utilizes the Simplex procedure of *Nelder and Mead* [1965]. After the CCE algorithm terminates, the entire population is recombined and then partitioned again into a number of complexes and this shuffling step functions to share information between complexes. This process is repeated many times and as the search process continues, the entire population tends to converge to a local or global optimum. See *Duan et al.* [1992, 1993, 1994] for a detailed description of SCE.

2.2. Dynamically Dimensioned Search Algorithm

[14] The DDS algorithm is a novel and simple stochastic single-solution based heuristic global search algorithm that was developed for the purpose of finding good global solutions (as opposed to globally optimal solutions) within a specified maximum function (or model) evaluation limit. The algorithm is designed to scale the search to the user-specified number of maximum function evaluations and thus has no other stopping criteria. In short, the algorithm searches globally at the start of the search and becomes a more local search as the number of iterations approaches the maximum allowable number of function evaluations. The adjustment from global to local search is achieved by dynamically and probabilistically reducing the number of dimensions in the neighborhood (i.e., the set of decision variables or parameters modified from their best value). The decision variables in automatic calibration are the model parameters being calibrated, and the dimension being varied is the number of model parameter values being changed to generate a new search neighborhood. Candidate solutions are created by perturbing the current solution values in the randomly selected dimensions only. These perturbations magnitudes are randomly sampled from a normal distribution with a mean of zero. Our algorithm design choices to select a subset of dimensions for perturbation completely at random without reference to sensitivity information and the use of the normal distribution were made to keep DDS as simple and parsimonious as possible. However, DDS could also be applied with alternative probability distributions. DDS is a greedy type of algorithm since the current solution, also the best solution identified so far, is never updated with a solution that has an inferior value of the objective function. The complete DDS algorithm pseudocode for minimization is provided in Figure 1.

[15] The DDS algorithm is unique relative to current optimization algorithms because of the way the neighborhood is dynamically adjusted by changing the dimension of the search. For example, the dynamic adjustment in the number of parameter dimensions varied in the neighborhood (step 3 in Figure 1) distinguishes DDS from adaptive random sampling (ARS) as described by *Masri et al.* [1980], the ARS implementation used by *Duan et al.* [1992] and a (1 + 1) evolutionary strategy (ES). *Masri et al.* [1980] adjust neighborhood size by modifying the perturbation magnitude (variance) in each dimension while *Duan et al.* [1992] narrow the sampling bounds. In a basic (1 + 1) ES with the 1/5th success rule for step length control [see, e.g., *Schwefel*, 1995], the mutation variances increase or decrease in response to whether the objective function has recently been improved. In contrast, the DDS perturbation variances remain constant and the number of decision variables perturbed from their current best value decreases as the number of function evaluations approaches the maximum function evaluation limit. This key feature of DDS was motivated by our experience with manual calibration of watershed models where early in the calibration exercise relatively poor solutions suggested the simultaneous modification of a number of model parameters but as the calibration results improved, it became necessary to only modify one or perhaps a few parameters simultaneously so that the current gain in calibration results were not lost.

[16] The only algorithm parameter to set in the DDS algorithm is the scalar neighborhood size perturbation parameter (r) that defines the random perturbation size standard deviation as a fraction of the decision variable range. A default value of the r parameter is recommended as

STEP 1. Define DDS inputs:
- neighborhood perturbation size parameter, r (0.2 is default)
- maximum # of function evaluations, m
- vectors of lower, $\boldsymbol{x}^{min}$, and upper, $\boldsymbol{x}^{max}$, bounds for all D decision variables
- initial solution, $\boldsymbol{x}^0 = [x_1, \ldots, x_D]$

STEP 2. Set counter to 1, i = 1, and evaluate objective function F at initial solution, $F(\boldsymbol{x}^0)$:
- $F_{best} = F(\boldsymbol{x}^0)$, and $\boldsymbol{x}^{best} = \boldsymbol{x}^0$

STEP 3. Randomly select J of the D decision variables for inclusion in neighborhood, $\{N\}$:
- calculate probability each decision variable is included in $\{N\}$ as a function of the current iteration count: $P(i) = 1 - \ln(i)/\ln(m)$
- FOR d = 1, … D decision variables, add d to $\{N\}$ with probability P
- IF $\{N\}$ empty, select one random d for $\{N\}$

STEP 4. FOR j = 1, …, J decision variables in $\{N\}$, perturb x_j^{best} using a standard normal random variable, N(0,1), reflecting at decision variable bounds if necessary:
- $x_j^{new} = x_j^{best} + \sigma_j N(0,1)$, where $\sigma_j = r(x_j^{max} - x_j^{min})$
- IF $x_j^{new} < x_j^{min}$, reflect perturbation:
  - $x_j^{new} = x_j^{min} + (x_j^{min} - x_j^{new})$
  - IF $x_j^{new} > x_j^{max}$, set $x_j^{new} = x_j^{min}$
- IF $x_j^{new} > x_j^{max}$, reflect perturbation:
  - $x_j^{new} = x_j^{max} - (x_j^{new} - x_j^{max})$
  - IF $x_j^{new} < x_j^{min}$, set $x_j^{new} = x_j^{max}$

STEP 5. Evaluate $F(\boldsymbol{x}^{new})$ and update current best solution if necessary:
- IF $F(\boldsymbol{x}^{new}) \leq F_{best}$, update new best solution:
  - $F_{best} = F(\boldsymbol{x}^{new})$ and $\boldsymbol{x}^{best} = \boldsymbol{x}^{new}$

STEP 6. Update iteration count, i = i+1, and check stopping criterion:
- IF i = m, STOP, print output (e.g. F_{best} & $\boldsymbol{x}^{best}$)
- ELSE go to STEP 3

Figure 1. Dynamically dimensioned search (DDS) algorithm.

0.2 (and used in this study) because this yields a sampling range that practically spans the normalized decision variable range for a current decision variable value halfway between the minimum and maximum. This sampling region size is designed to allow the algorithm to escape regions around poor local minima. An r value of 0.2 means that for a decision variable with a range of 10 units, the standard deviation of the perturbation random variable is equal to $0.2 \times 10 = 2$ units. Thus r is a scaling factor assigning the same relative variation to each decision variable (relative to the decision variable ranges). In cases where the initial solution is known to yield good objective function values, reducing r to perhaps 0.1 may also be reasonable to better focus the search around the initial solution. However, in all other cases, reducing r below 0.2 is only recommended to restart terminated DDS runs for which further solution refinement is desired.

[17] The one-dimensional decision variable perturbations in step 4 of the DDS algorithm (Figure 1) can generate new decision variable values outside of the decision variable bounds (or box constraints). In order to ensure that each one-dimensional perturbation results in a new decision variable that respects the bounds, the minimum and maximum decision variable limits act as reflecting boundaries in the DDS algorithm. (see step 4 of Figure 1.) For example, if a random perturbation went 0.2 units past the lower boundary, the new decision variable for the candidate solution would be the minimum value plus 0.2. This reflecting boundary approach allows decision variable values to more easily approach their minimum or maximum values in comparison with a simple perturbation resampling approach for ensuring decision variable boundaries are respected.

[18] The maximum number of function evaluations (m) is an algorithm input (like the initial solution) rather than algorithm parameter because it should be set according to the problem specific available (or desired) computational time to expend on the optimization problem. The value of m therefore depends on the time to compute the objective function and the available computational resources. Except for the most trivial objective functions, essentially 100% of DDS execution time is associated with the objective function evaluation. Remember that the DDS algorithm scales the search strategy from global in the initial iterations to more local in the final iterations regardless of whether m is 100 or 10,000 function evaluations. After initial testing, it was decided that in the absence of a specific initial solution, a simple approach to reduce DDS sensitivity to a poor randomly sampled initial solution was to initialize DDS to the best of M uniform random solutions, where M is the largest integer of 0.005m and 5. It must be clarified that the DDS algorithm is not designed to converge to the precise global optimum. Instead, it is designed to converge to the region of the global optimum in the best case or the region of a good local optimum in the worst case. Local, perhaps derivative-based, searches could be initialized from the final DDS solution in order to identify a more precise estimate of the local optimum close to where DDS converged. However, this additional solution refinement step may be unnecessary in many practical model calibration case studies.

2.3. Optimization Test Functions and Problems

[19] Four difficult generalized D-dimensional global optimization test functions (Rastrigin, Griewank, Ackley and Bump) were selected in order to compare SCE and DDS performance. As global optimization test functions, they each have a high number of local optima, which is a type of problem both SCE and DDS are designed to solve. Table 1 lists the functions and their characteristics (see *Tolson* [2005] for Bump function). Ten and 30 dimensions were selected for the Rastrigin, Griewank and Ackley functions to roughly span the range of dimensions for the real SWAT2000 model calibration problems defined later in section 2.4. The Bump function is a maximization problem with 20 dimensions from *Keane* [1996]. For the Bump function and all other maximization problems, the optimization algorithms minimize (-1) times the objective function value.

Table 1. Summary of Optimization Test Functions

Reference/Name	Equation	Dimensions D	Bound and Other Constraints	Minimum
Rastrigin [1974]	$f(\mathbf{x}) = \sum_{i=1}^{D}[x_i^2 - \cos(2\pi x_i)]$	10 and 30	$[-2, 2]^D$	-D
Griewank [1981]	$f(\mathbf{x}) = \sum_{i=1}^{D}(x_i^2/4000) - \prod_{i=1}^{D}\cos(x_i/\sqrt{i}) + 1$	10 and 30	$[-500, 700]^D$	0
Ackley [1987]	$f(\mathbf{x}) = -20\exp\left[\sqrt{\frac{1}{D}\sum_{i=1}^{D}x_i^2}\right] - \exp\left[\frac{1}{D}\sum_{i=1}^{D}(\cos(2\pi x_i))\right]$	10 and 30	$[-1, 3]^D$	-20-e
Sixpar	see *Duan et al.* [1992]	6	*Duan et al.* [1992]	0

[20] The SCE algorithm was originally demonstrated as an effective and efficient algorithm based largely on a synthetic six-dimensional hydrologic model calibration problem called 'Sixpar' [*Duan et al.*, 1993, 1992]. Therefore it was deemed important to revisit the Sixpar test problem in this study. Hence Matlab SCE, DDS, and Fmincon algorithms were applied to the Sixpar problem. The objective for the Sixpar problem is to minimize the sum of squared errors (SSE) between the model predictions and a synthetically generated time series of measured data. *Duan et al.* [1993] report all other details on the problem except that in addition to the bound constraints on the six decision variables, there are four linear constraints that are checked in the Sixpar example. The original SCE algorithm handles constraint violations by randomly sampling around the best solution when one or more of the constraints (bound or linear) are violated. The Matlab SCE handles constraint violations the same way as SCE-UA. Linear constraint handling for the Sixpar problem in this study is discussed further in section 3.4.

2.4. SWAT2000 Cannonsville Watershed Model Calibration Case Study

[21] *Tolson and Shoemaker* [2004] and *Tolson* [2005] recently applied a slightly modified version of the SWAT2000 [*Neitsch et al.*, 2001] watershed simulation model to predict flow, sediment and phosphorus delivery to the Cannonsville Reservoir in upstate New York. The nearly 1200 km^2 reservoir watershed is dominated by forests and agricultural lands and is only about 1% urban land. *Tolson* [2005] manually calibrated the model to a rich data set for flow, total suspended sediment (TSS) and phosphorus. Multiple SWAT2000 model calibration problems were derived from the Cannonsville case study and formulated as optimization problems and then solved with the DDS and the other algorithms listed in section 2.1.

[22] The Soil and Water Assessment Tool version 2000 (SWAT2000) is a spatially distributed continuous simulation model for predicting flow, sediment, nutrient and other contaminant transport. SWAT2000 is designed to compute long-term runoff and nutrient export from rural watersheds, especially those dominated by agriculture [*Arnold et al.*, 1998] like the Cannonsville Reservoir Watershed. The model is maintained by the Agricultural Research Service of the U.S. Department of Agriculture (USDA) and distributed by the U.S. Environmental Protection Agency (EPA) for nonpoint source modeling. Further details about the SWAT2000 model application to the Cannonsville Reservoir watershed are provided by *Tolson* [2005] and *Tolson and Shoemaker* [2004].

[23] The SWAT2000 model Fortran source code was slightly modified and then recompiled in order to create an efficient case study. SWAT2000 was changed so that all model optimization parameters were read from the input files to five decimal places of precision and unnecessary output files and screen printing were eliminated. Since Matlab was the programming language used to code all optimization algorithms, Matlab programs were mainly used to transfer new model parameter values to the appropriate model input files and extract model time series predictions from the model output files and then calculate the necessary objective function values.

[24] Two scales of SWAT2000 models within the Cannonsville Reservoir Watershed were utilized in this study. The multiple subbasin watershed model (43 subbasins, 758 hydrologic response units) was used here to calibrate model predictions to the main watershed monitoring stations (Walton and Beerston). The largest flow monitoring location in the watershed is the Walton USGS station (01423000) located on the main stem West Branch Delaware River and drains an area of 860 km^2. The Beerston water quality station is just downstream of Walton and drains 913 km^2. The Walton/Beerston calibration was the main focus of the manual calibration effort in *Tolson* [2005] and is therefore repeated here. However, since one execution of the Cannonsville watershed scale model requires 2 minutes of computation time on a Pentium IV 3 GHz processor even after substantial code optimizations, the total number of model evaluations available for calibration was quite limited. This is especially true since replicate optimization trials were implemented to compare multiple algorithms. Therefore a second single subbasin model was created for the Town Brook subwatershed (37 km^2 drainage area) to allow for a higher number of model evaluations in algorithm testing. Town Brook is also monitored for flow (USGS station 01421618) and water quality. The New York State Department of Environmental Conservation (NYSDEC) provided daily TSS and total phosphorus loads calculated at each monitoring location.

[25] A number of SWAT2000 model parameters optimized in this case study are not spatially variable and are instead a constant value across all model spatial units (e.g., all snowmelt parameters). Other parameters such as SCS curve numbers and soil properties are spatially variable and can therefore be assigned different values for different spatial units. These could be calibrated by considering each spatial unit parameter value an independent calibration parameter. However, in this study, such an approach would have increased the total number of calibrated parameters to more than 100. Since the calibration formulations presented

Table 2. SWAT2000 Flow-Related Parameters Optimized in Formulations 1, 2, and 3

Parameter	Brief Description (units)	Minimum	Maximum
SFTMP	snow fall temperature (°C)	−5	5
SMTMP	snowmelt temperature threshold (°C)	−5	5
SMFMX	melt factor for snow (mm H_2O/°C day)	1.5	8
TIMP	snowpack temperature lag factor	0.01	1
SURLAG	surface runoff lag coefficient	1	24
GW_DELAY	groundwater delay time (days)	0.001	500
ALPHA_BF	base flow alpha factor	0.001	1
GWQMN	threshold groundwater depth for return flow (mm)	0.001	500
LAT_TIME	lateral flow traveltime (days)	0.001	180
ESCO	soil evaporation compensation factor	0.01	1
CN2_f[a]	runoff curve number multiplicative factor	0.75	1.25
AWC_f[b]	available water capacity range factor	0	1
Ksat_f[b]	saturated hydraulic conductivity range factor	0	1
DepthT_f[b]	soil profile total depth range factor	0	1

[a]CN2_f is a multiplicative factor used to simultaneously adjust all spatially variable base runoff curve numbers (CN2) up to a maximum 98.0.
[b]AWC_f, Ksat_f, and DepthT_f are factors linearly scaling the physical properties (AWC, Ksat, and DepthT) between their minimum (factor = 0) and maximum (factor = 1) soil type specific values. The ranges of AWC and Ksat were derived from soil survey data while the total soil depth range (DepthT) range was assumed to be ±25% of the soil survey data.

below focus on one monitoring location and thus represented the integral of all upstream spatial predictions, spatially variable model parameters were not calibrated for each spatial unit independently. Instead, a single calibration factor was used to increase or decrease spatially variable parameter values from their base or default values. For each spatially variable parameter, this approach maintained the relative differences in the base or default parameter values assigned to different spatial units.

2.4.1. Flow Calibration Formulations

[26] Three flow calibration problems were formulated based on the set of SWAT2000 flow parameters calibrated manually by *Tolson* [2005]. These 14 model parameters impact snowmelt, surface runoff, groundwater, lateral flow and evapotranspiration predictions and are listed along with their ranges in Table 2. The parameter ranges were based mainly on ranges in the SWAT2000 model documentation [*Neitsch et al.*, 2001] in order to replicate the automatic calibration process that would have occurred with little prior knowledge of the model application to this case study area. For computational efficiency reasons, the model calibration formulations below are based on model simulation time periods that were shorter than the calibration periods used by *Tolson* [2005]. Therefore automatic flow calibration results achieved here are not compared to manual calibration results of *Tolson* [2005].

2.4.1.1. Formulation 1

[27] The 37 km^2 Town Brook single subbasin SWAT2000 model was calibrated for flow against real measured flow data according to the optimization model below:

$$\min_{\mathbf{x}} \ \mathrm{SSE}^{\mathrm{Q}}(\mathbf{x}) = \sum_{t=1}^{T} (Qmeas_t - Qsim_t)^2$$
$$\text{s.t.} \ \ x_d^{\min} \le x_d \le x_d^{\max}, d = 1, \ldots, D \qquad (1)$$

where $\mathrm{SSE}^{\mathrm{Q}}$ is the sum of squared error for daily flows, $\mathbf{x}$ is a vector of D model parameters that are each subject to bound constraints listed in Table 2, $Qmeas_t$ and $Qsim_t$ are the measured and simulated flows on day t and T is the total number of days in the calibration period. For this formulation, D is 14 and T is 1096 days (October 1997 to September 2000). The best theoretical $\mathrm{SSE}^{\mathrm{Q}}$ value is 0.0. However, the true minimum $\mathrm{SSE}^{\mathrm{Q}}$ for this real calibration problem, as with all others, is some unknown positive quantity.

2.4.1.2. Formulation 2

[28] The Town Brook SWAT2000 model used in Formulation 1 was calibrated for flow against synthetically generated flow data. The optimization model in formulation 2 is exactly the same as equation (1) except that the measured data (*Qmeas*) was synthetically generated using the manually calibrated SWAT2000 model parameter values established by *Tolson* [2005]. Therefore the known solution to this problem has a minimum $\mathrm{SSE}^{\mathrm{Q}}$ of 0.0.

2.4.1.3. Formulation 3

[29] The 1200 km^2 Cannonsville Reservoir multiple subbasin SWAT2000 model was calibrated for flow against real measured flow data according to the following:

$$\max_{\mathbf{x}} \ \mathrm{E}_{\mathrm{NS}}^{\mathrm{Q}}(\mathbf{x}) = 1 - \frac{\sum_{t=1}^{T} (Qmeas_t - Qsim_t)^2}{\sum_{t=1}^{T} \left(Qmeas_t - T^{-1} \sum_{t=1}^{T} Qmeas_t\right)^2}$$
$$\text{s.t.} \ x_d^{\min} \le x_d \le x_d^{\max}, d = 1, \ldots, D \qquad (2)$$

where $\mathrm{E}_{\mathrm{NS}}^{\mathrm{Q}}$ is the Nash-Sutcliffe coefficient for daily flows (Q), $\mathbf{x}$ is a vector of D model parameters that are each subject to bound constraints listed in Table 2, $Qmeas_t$ and $Qsim_t$ are the measured and simulated flows on day t and T is the total number of days in the calibration period. E_{NS} values range from negative infinity to 1. The true maximum $\mathrm{E}_{\mathrm{NS}}^{\mathrm{Q}}$ value in this case study, as with all others, is not known. For this formulation, D is 14 and T is 2191 days (1990–1995). As described for the Bump function in section 2.3, this and all other maximization problems are solved by minimizing (−1) times the objective function value.

2.4.2. Simultaneous Flow, Sediment, and Phosphorus Calibration Formulations

[30] Two calibration problems were formulated for the simultaneous calibration of flow, sediment and phosphorus. The model was simultaneously calibrated to the three constituents since flow predictions, namely surface runoff volumes, largely influence the water quality predictions. The parameters to be optimized were selected based on the set of model parameters that *Tolson* [2005] modified from their default values. These 30 optimized model parameters

Table 3. SWAT2000 Flow-, Sediment-, and Phosphorus-Related Parameters Optimized in Formulations 4 and 5

Parameter	Brief Description (units)	Minimum	Maximum
See Table 2	same parameters and ranges as formulations 1, 2 and 3	-	-
APM	tributary channel peak rate adjustment sediment routing factor	0.5	1.5
PRF_a	main channel peak rate adjustment sediment routing factor	0.5	1.5
SPCON[a]	channel sediment routing parameter (linear)	0.0001	0.001
SPEXP[a]	channel sediment routing parameter (exponential)	1	2
PPERCO	phosphorus (P) percolation coefficient (10 m^3/Mg)	10	17.5
PHOSKD	P soil partitioning coefficient (m^3/Mg)	100	200
CMN	rate factor for humus mineralization of active organic P	0.0001	0.003
UBP	plant P uptake distribution parameter	0.1	100
LAT_SED	sediment concentration in lateral & groundwater flow (mg/L)	0.1	22.8
ERGORGP	P enrichment ratio for loading with sediment	1	5
SLSUBBSN_f[b]	average slope length (m)	0.5	1.5
SLSOIL_f[b]	slope length for lateral subsurface flow (m)	0.5	1.5
CH_EROD[a]	channel erodibility factor	0	0.6
CLAY_f[c]	soil layer clay content range factor	0	1
ROCK_f[c]	soil layer rock content range factor	0	1
MUSLEadj[d]	erosion under snow cover adjustment parameter	0	1

[a]Parameters are not used in Town Brook model (formulation 4) because the processes the parameters control are not simulated for a single subbasin model.

[b]SLSUBBSN_f, SLSOIL_f, and CN2_f are multiplicative factors used to simultaneously adjust all spatially variable base values of the SLSUBBSN, SLSOIL, and CN2 parameters, respectively.

[c]CLAY_f and ROCK_f are factors linearly scaling the physical properties (CLAY and ROCK) between their minimum (factor = 0) and maximum (factor = 1) soil type specific values. The ranges for CLAY and ROCK were derived from soil survey data.

[d]The MUSLEadj parameter was added to SWAT2000 for the Cannonsville application of *Tolson* [2005] and controls the snow cover influence on hydrologic response unit sediment yield.

include the 14 flow parameters considered in formulations 1, 2 and 3. In addition, land surface and channel erosion parameters, as well as phosphorus related parameters were considered as optimization parameters. The optimized parameter names and ranges are listed in Table 3. The parameter ranges were based mainly on ranges in the SWAT2000 model documentation [*Neitsch et al.*, 2001] in order to replicate the automatic calibration process that would have occurred with little prior knowledge of the model application to this case study area.

2.4.2.1. Formulation 4

[31] The 37 km^2 Town Brook SWAT2000 model used in formulation 1 was calibrated simultaneously for flow, sediment and total phosphorus against real daily flow and water quality loading data according to the following:

$$\begin{aligned}\max_{\mathbf{x}}\ E_W(\mathbf{x}) = {} & 0.5\left[E_{NS}^{Q} - \max\left(0, |\%B^{Q}| - 10\right)*0.01\right] \\ & + 0.2\left[E_{NS}^{TSS} - \max\left(0, |\%B^{TSS}| - 30\right)*0.01\right] \\ & + 0.3\left[E_{NS}^{TP} - \max\left(0, |\%B^{TP}| - 30\right)*0.01\right] \\ & \text{s.t. } x_d^{\min} \le x_d \le x_d^{\max}, d = 1, \ldots, D \end{aligned} \quad (3)$$

where E_w is a weighted summation of reduced Nash-Sutcliffe coefficients for flow (E_{NS}^{Q}), total suspended sediment (E_{NS}^{TSS}) and total phosphorus (E_{NS}^{TP}), $\mathbf{x}$ is a vector of D model parameters that are each subject to bound constraints in Table 3, and %B is the percent bias of model predictions for a given constituent calculated as follows:

$$\%B = \frac{100\left(\sum_{t=1}^{T} \text{Simulated}_t - \sum_{t=1}^{T} \text{Measured}_t\right)}{\sum_{t=1}^{T} \text{Measured}_t} \quad (4)$$

[32] The weights assigned to equation (3), which are 0.5 for flow, 0.2 for TSS and 0.3 for TP, reflect the higher quality and longer period of flow data (October 1997 to September 2000) relative to water quality data (only October 1998 to September 2000) and the fact that accurate phosphorus prediction is more important than accurate TSS prediction in this case study. Equation (3) combines the %B and E_{NS} values into a single objective function designed to maximize E_{NS} and reduce |%B| values to a specific threshold. Minimizing |%B| was only considered beneficial up to these thresholds (10% for flow, 30% for TP and TSS) due to the data errors associated with large flow events. E_w, like a real E_{NS} coefficient, ranges from negative infinity to 1. However, the true maximum E_w value is not known. For this formulation, D is 26 and T is 1096 days for flow and 731 days for TSS and TP.

2.4.2.2. Formulation 5

[33] The 1200 km^2 Cannonsville Reservoir Watershed multiple subbasin SWAT2000 model used in formulation 2 was calibrated simultaneously for flow, sediment and total phosphorus against real daily flow data at Walton and real daily water quality loading data at Beerston. The optimization model has the same form as equation (3). The only differences are that since data were deemed to be slightly more reliable for Beerston, the thresholds for applying a %Bias penalty for TSS and TP were reduced to 20% (from 30%), the number of calibrated parameters increases to 30, and T is 2191 days for flow and 1553 days for TSS and TP.

[34] There are a myriad of alternative ways to formulate the above calibration problems including solving the calibration problem as a multiobjective problem. The calibration problems formulated here are relatively simple and could be further extended to consider alternative low-flow weighting schemes, alternative performance statistics, and even take better advantage of other available measured data sources within the basin. However, for the purposes of

demonstrating the single-objective DDS optimization algorithm and comparing it to the single-objective SCE optimization algorithm, this set of calibration formulations was deemed sufficient.

2.5. Outline of Algorithm Comparisons

[35] Since this study is focused on the introduction of the DDS algorithm and the subsequent comparison to SCE with respect to optimization performance for model calibration problems, these comparisons are focused on the ability of each algorithm to optimize the value of the objective function in both the test function and calibration formulation problems considered here. Results are evaluated from a distributed model calibration perspective such that total function (i.e., model) evaluations are limited and thus achieving a highly precise globally optimal solution is an unreasonable expectation. In addition, errors in measured data undermine the value of such precision in parameter calibration. Given this perspective and the fact that we do not know the optimal solution in the real calibration formulations, we do not report on the final parameter values returned by DDS or SCE.

[36] Due to the stochastic nature of the SCE and DDS algorithm, their relative performance must be assessed over multiple optimization trials each initialized to independent populations or solutions. Algorithms are compared using 5 to 100 optimization trials. Optimization test function comparisons cover 10-, 20-, and 30-dimensional (i.e., the number of decision variables) problems while comparisons on the watershed model calibration formulations are presented for 6-, 10-, 14-, 26-, and 30-dimensional problems (i.e., the number of calibrated model parameters). The maximum number of model or function evaluations per optimization trial varies from 1000 to 10,000 (except for one test function). Since it is typical in global optimization to compare average algorithm performance [*Ali et al.*, 2005], average algorithm convergence in terms of the best solution found is plotted against the number of objective function evaluations for each algorithm. In other words, for a particular algorithm, the average of the best solution found so far across all optimization trials is computed after each additional objective function evaluation. These algorithm convergence plots provide comparative results relevant to future modelers who are constrained to anywhere from $\sim$100 to $\sim$10,000 model evaluations for calibration. Given that average algorithm performance does not provide a complete picture of results, the distribution or range of the best DDS and SCE objective function values is also graphically assessed for the majority of optimization problems considered.

[37] Unless otherwise noted, the initial solution for DDS is generated as described in section 2.2 and the neighborhood size parameter, r, is set to the default value of 0.2. All Fmincon and Simplex searches are initialized to the same set of initial solutions that start the DDS algorithm. Stopping criteria for the Fmincon and Simplex algorithms are noted in section 3 for each problem they are applied to. All SCE results presented in this paper are based on our Matlab SCE algorithm unless otherwise noted. As in the original SCE-UA algorithm, our SCE is initialized to a population generated by uniform random sampling. Default SCE algorithm parameters recommended by *Duan et al.* [1994] are used except in some problems the number of complexes was tuned to improve SCE results. The use of default algorithm parameters best replicates how the majority of modelers would use each algorithm. SCE and DDS are only stopped when the maximum function evaluation limit is reached.

3. Results

[38] The results are presented here in four sections. Section 3.1 compares the DDS algorithm with the SCE algorithm for a few common optimization test functions. In section 3.2, the algorithm results for the SWAT2000 Cannonsville Watershed real and synthetic flow calibration formulations for Town Brook and Walton are presented. Section 3.3 compares SCE and DDS performance on the high-dimension SWAT2000 Cannonsville Watershed simultaneous flow, sediment and phosphorus real calibration problems for Town Brook and Walton. Lastly, section 3.4 revisits the original synthetic Sixpar hydrologic model calibration problem as described and investigated by *Duan et al.* [1992, 1993].

3.1. Algorithm Comparisons for the Test Functions

[39] The average SCE and DDS algorithm results for some of the test functions listed in Table 1 are shown in Figure 2 as a function of the number of function evaluations. The number of SCE complexes for the 10-D test functions was set to four based on recommendations by *Duan et al.* [1994] for 10-D optimization problems. *van Griensven and Bauwens* [2003] used SCE with five complexes and approximately 8000 function evaluations for a 32-D model calibration problem. Therefore it was assumed reasonable to use five complexes for the 30-D test functions optimized with a maximum of 10,000 function evaluations. The 30-D test functions were also minimized using 10 complexes based on other studies that report using a much higher number of complexes [e.g., *Franchini et al.*, 1998; *Kuczera*, 1997]. The averages for the 10-D test functions are for 100 optimization trials while the averages for the 30-D test functions are for 30 optimization trials. For each trial, we record the best solution found on or before the ith function evaluation for a specific algorithm and then average these best solutions over the total number of trials for that algorithm to obtain the "average best function value" for i function evaluations. The maximum number of function evaluations used for the test functions (1000–10,000) were fixed so as to cover the range of maximum SWAT model evaluations used to solve the calibration case study formulations in sections 3.2 and 3.3. Algorithm comparisons for the test functions in Figure 2 were limited to SCE and DDS in order to clearly highlight some general algorithm performance differences.

[40] In general, Figure 2 shows that DDS does better if the number of function evaluations is limited. For the Griewank 10-D function, the SCE algorithm finds slightly better average solutions than DDS after 2000 function evaluations. However, DDS provides substantially better average solutions than SCE for the Griewank 10-D function when function evaluations are limited to fewer than around 1000 function evaluations. Similar results are obtained (although not presented) for the Ackley 10-D function and the Griewank 30-D function. Figure 2c shows a comparable pattern for the Ackley 30-D function in that the SCE

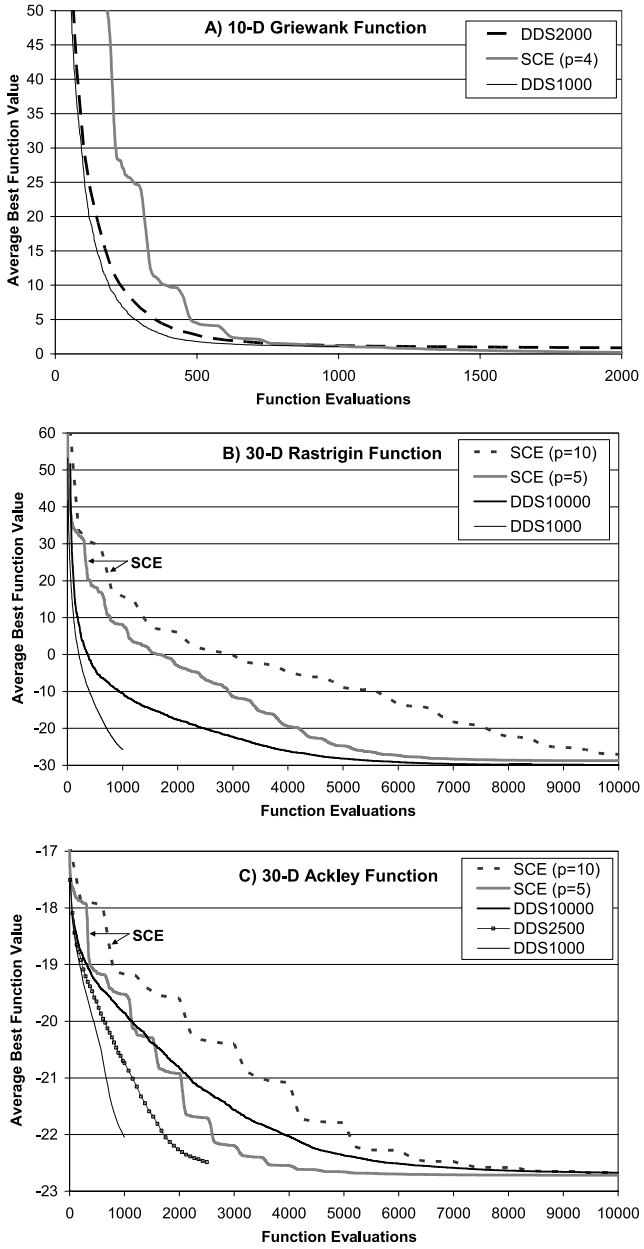

Figure 2. SCE (with varying complexes, p) and DDS (with varying maximum function evaluation limits) performance comparisons for various 10-D and 30-D optimization test functions: (a) 10-D Griewank, (b) 30-D Rastrigin, and (c) 30-D Ackley. In the legend the DDS number means *m* set to that number of function evaluations; for example, DDS2000 indicates *m* set to 2000 function evaluations.

algorithm finds slightly better average solutions than DDS after 10,000 function evaluations. However, DDS provides substantially better average solutions than SCE for the Ackley 30-D function in Figure 2c when function evaluations are more limited (less than ~3000). This substantial performance difference is clearly demonstrated in Figure 2c by the DDS optimization runs to a maximum of 1000 and 2500 function evaluations. Unlike the other two functions in Figure 2, DDS results for the Rastrigin 30-D (Figure 2b) are better than SCE across all function evaluations considered.

[41] Results from Figure 2 show that SCE is sensitive to the number of complexes selected. To investigate this more closely and try to determine the optimal number of SCE complexes for a specific example, the 10-D Rastrigin function was minimized with SCE using 1, 2, 3 and 4 complexes and results are compared to the default DDS results in Figure 3. Figure 3a shows the average results and identifies an Ideal SCE series that is the best average results across all four SCE parameter settings. The Ideal SCE series is clearly unattainable with a single SCE optimization run and shows that the optimal number of complexes depends very much on the number of function evaluations utilized. For example, SCE (p = 1) is best for less than about 600 function evaluations but SCE (p = 2) is then best for between 600 to 1000 function evaluations. In contrast, DDS2000 produces lower function values on average than the Ideal SCE for nearly all function evaluations (except between 250 and 500) and the DDS1000 results are even lower than DDS2000.

[42] Figure 3b shows the empirical cumulative distribution function of the final best objective function values from all 100 optimization trials after 2000 function evaluations. The minimum function value is −10 and the almost vertical line at −10 for DDS indicates all of the DDS solutions are very close (within 0.08) to the minimum. The vertical lines for SCE at −9.0, −8.0, −7.0, and −6.0 indicate SCE converges to very poor local minima with significant to substantial frequency (5–75% of trials) depending on the number of complexes. Note that the SCE results for one and two complexes typically converge in fewer than 1000 function evaluations and thus could be improved by simply restarting SCE and using a total of 2000 total function evaluations. Even with one or two restarts, SCE with one or two complexes would still converge to quite poor solutions with notable frequency. Although not obvious due to scaling in the figure, when SCE does manage to avoid the poor local minima, it converges to the global minimum with more significant digits than DDS. However, improving the final DDS solutions after 2000 function evaluations with the derivative-based Fmincon algorithm also yielded the global minimum in all 100 trials with less than 100 additional function evaluations on average.

[43] The DDS and SCE algorithms were also tested on a 20-D version of the extremely multimodal "bump" function [*Keane*, 1996]. The bump function has one nonlinear and one linear constraint in addition to decision variable bound constraints. Following the approach taken by *Keane* [1996], all SCE and DDS solutions that were infeasible were simply assigned a function value of 0. In comparison with five other global optimization methods (including genetic algorithms and evolution strategy), DDS generated the best average results after 150,000 function evaluations (slightly better than the GA) while SCE performed second worst out of six algorithms on this problem. Complete Bump function results are available from *Tolson* [2005].

3.2. SWAT2000 Flow Calibration

[44] The SWAT2000 flow calibration problems described in section 2.4.1 are solved here using the optimization algorithms described in section 2.1. The default number of SCE complexes (*p*) for the three 14-D flow calibration problems was set to four based on recommendations by *Duan et al.* [1994] for a 13-D calibration problem.

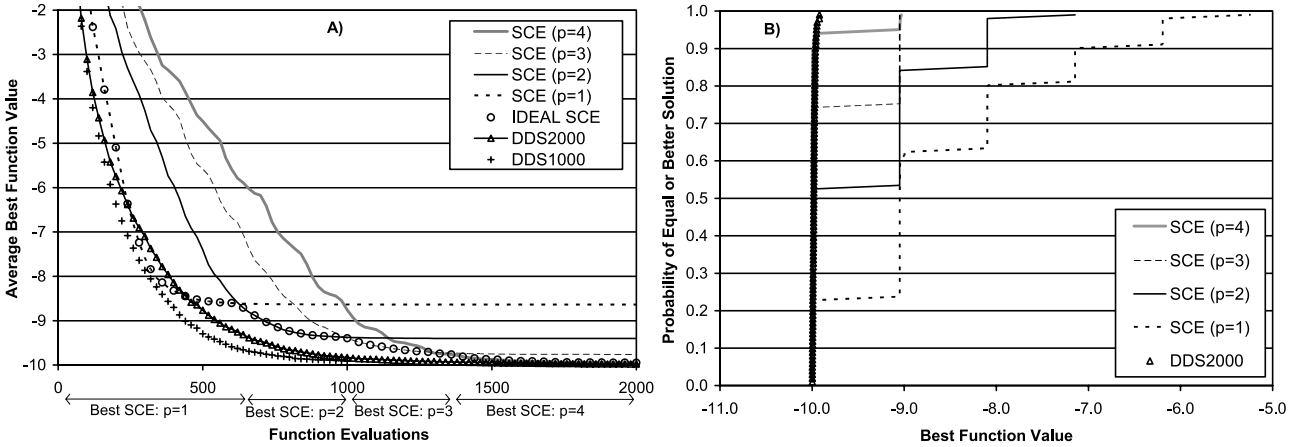

Figure 3. SCE and DDS performance comparisons based 100 optimization trials of the 10-D Rastrigin function: (a) Average best function values and (b) empirical cumulative distribution function of final best function values after a maximum of 2000 function evaluations. The ideal SCE series is the unattainable best average SCE result of the four SCE parameter settings considered.

[45] Figure 4 summarizes algorithm performance on formulation 1. The DDS, SCE with two and four complexes, Simplex and Fmincon results are plotted versus the number of model evaluations in Figure 4a. Fmincon and Simplex optimization trials used two restarts and both results in Figure 4 are shown as points connected by a dashed line because intermediate algorithm results were unavailable and were therefore linearly interpolated. The Fmincon and Simplex points in Figure 4 are the average function values plotted against the corresponding average number of model evaluations required for convergence after one and two restarts. The minimum difference for computing finite difference derivatives in Fmincon was set to 10^{-5} to match the modified SWAT2000 input precision levels. All other Fmincon and Simplex algorithm inputs were left at their default values. Clearly, both Fmincon and the Simplex are converging to poor local minima with the simplex performing slightly better. The default SCE (p = 4) converges substantially slower than DDS and somewhat slower than SCE with p = 2. DDS finds lower average SSE values than either SCE result for any number of model evaluations in Figure 4a.

[46] Figure 4b provides a more complete description of algorithm performance by plotting all 20 objective function values for the DDS and SCE algorithms for fixed levels of computational effort (1000 and 2500 model evaluations). After 1000 model evaluations, the median DDS solution is lower than both SCE results. Reducing p to 2 improves average SCE performance after 1000 evaluations, but after 2500 evaluations, average and median SCE results are nearly indistinguishable. At 2500 model evaluations, Figure 4b shows that reducing the number of complexes increases the variance (i.e., spread) of SCE solutions. Although the variance of DDS solutions after 2500 model evaluations is noticeably higher than SCE with p = 4, half the DDS solutions are within 10 SSE units of the minimum SSE found (601).

[47] The synthetic Town Brook flow calibration (formulation 2) results for SCE, DDS, Fmincon and the Simplex algorithms are compared in Figure 5. The average best solutions are plotted on a logarithmic scale versus the number of SWAT model evaluations in order to clearly display the differences in algorithm performance. All Fmincon and Simplex algorithm inputs are the same as in formulation 1. Similar to Figure 4, Fmincon and Simplex

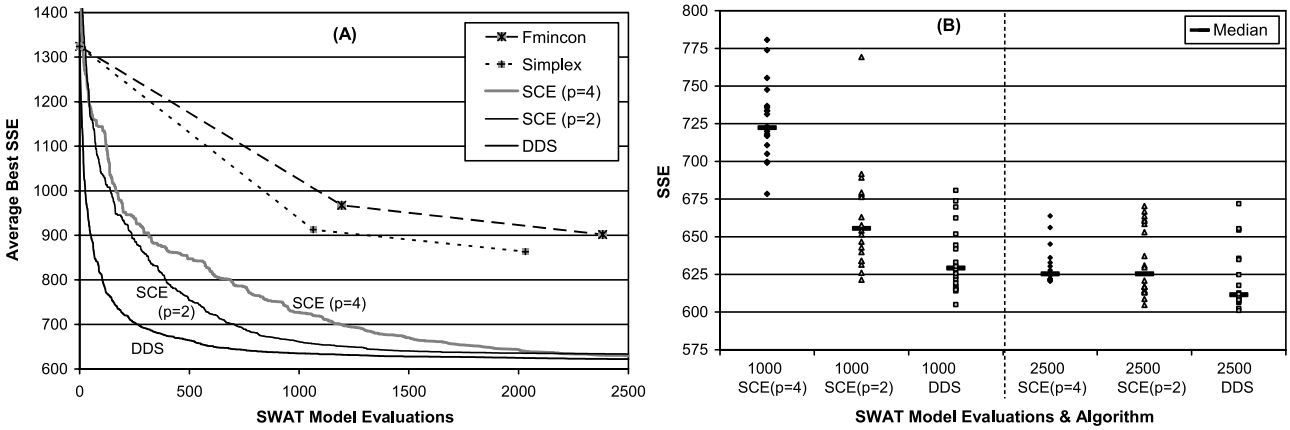

Figure 4. Algorithm performance comparisons for the 14-D Town Brook real flow calibration (formulation 1): (a) Average best SSE across 20 optimization trials as a function of the number of model evaluations. (b) All SCE and DDS solutions after 1000 and 2500 model evaluations.

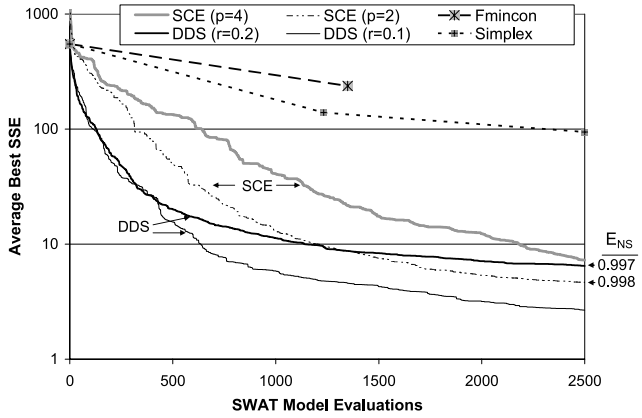

Figure 5. Algorithm performance comparison for the 14-D Town Brook synthetic flow calibration (formulation 2): Average best SSE across 20 (SCE p = 4, DDS r = 0.2, Fmincon, and Simplex) or 10 (SCE p = 2 and DDS r = 0.1) optimization trials as a function of the number of model evaluations.

converge to poor local solutions. Fmincon was not restarted since it was clear results would not substantially improve. Although the DDS and SCE algorithms converge to nearly the same SSE value after 2500 model evaluations under default algorithm parameters, the DDS algorithm reached good solutions more quickly. When p was reduced to two from four, SCE convergence speed and effectiveness was improved. In fact, SCE results with p = 2 are slightly better on average than the default DDS algorithm (r = 0.2) results after 2500 function evaluations. However, this difference is not significant from a practical calibration perspective since SCE improves the corresponding average E_{NS} coefficient by only 0.001 over DDS. If DDS is also fine-tuned by reducing r to 0.1, DDS produces the lowest average SSE values of all algorithms after 2500 model evaluations.

[48] The more computationally expensive Cannonsville Reservoir watershed calibration problem (formulation 3) was solved using only DDS and SCE algorithms since previous results for formulations 1 and 2 suggest that even with restarts, the Fmincon and Simplex algorithms were not capable of avoiding poor local minima. All 10 optimization trial results for this maximization problem are shown for both DDS and SCE in Figure 6 for up to 1000 SWAT model evaluations. The DDS algorithm clearly outperforms SCE for this problem for all model evaluations. Furthermore, the DDS variance is smaller than the variance in SCE solutions.

3.3. SWAT2000 Simultaneous Flow, Sediment, and Phosphorus Calibration

[49] The SWAT2000 simultaneous flow, TSS and phosphorus problems discussed in section 2.4.2 are solved here using only the SCE and DDS algorithms. Since the dimensionality and thus problem difficulty increases relative to the previous three flow calibration problems, the number of model evaluations used for calibration was also increased. The objective functions for formulations 4 and 5 can be roughly interpreted as a weighted E_{NS} coefficient.

[50] For the 26-D Town Brook calibration problem (formulation 4), 10,000 model evaluations were used for SCE while only 5000 model evaluations were used for DDS due to restrictions on available computational time. Five SCE complexes were selected based on *van Griensven and Bauwens* [2003]. In addition, an alternative SCE parameter setting approach described by *Sorooshian et al.* [1993] as relatively more efficient was evaluated where the number of complexes was set to five initially and then reduced to a minimum of two as the search progressed. Figure 7a shows the average best solution across 10 optimization trials for SCE and DDS versus the number of model evaluations. Results show the dominant performance of DDS over both SCE parameterizations. Even after twice as many model evaluations (10,000), the SCE (p = 5) average solution is still worse than DDS by more than 0.02 objective function units. Figure 7b presents all DDS solutions after 1000 and 5000 model evaluations and all SCE solutions after 5000 and 10,000 model evaluations. Although the SCE complex reduction strategy (p = 5 to p = 2) shows improved efficiency relative to SCE with a constant number of complexes, DDS results are still notably better than either SCE result. For example, Figure 7b shows the worst DDS solution after 5000 evaluations is approximately the same as the median SCE solution for either SCE parameterization.

[51] For the 30-D Cannonsville Reservoir Watershed problem (formulation 5), 2000 model evaluations were used for both SCE and DDS. Note that each optimization trial required approximately 68 hours of computation time on a Pentium IV, 3-Ghz processor. In light of the previous SCE results, SCE was applied to this problem using only two complexes in an attempt to improve SCE efficiency for this high-dimension, limited number of model evaluations problem. Figure 8 shows DDS and SCE average performance as well as the best and worst performance over five optimization trials. SCE simply cannot compete with DDS on this computational scale. The worst DDS solution was always better than the best SCE solution after approximately 200 function evaluations.

[52] Performance statistics for flow, TSS and total phosphorus are summarized in Table 4 for the best solutions from SCE and DDS to the simultaneous flow and water quality calibration problems (formulations 4 and 5). Disaggregating the objective function component performance statistics in equation (3), namely for flow, TSS and total P, demonstrates the differences in algorithm results in more interpretable units. For formulation 4, with only half the

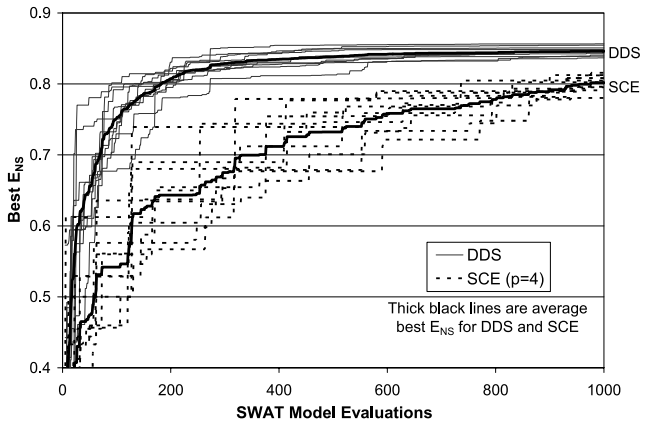

Figure 6. DDS and SCE results for 14-D Walton flow calibration (formulation 3) showing the distribution of best Nash-Sutcliffe coefficients (E_{NS}) for 10 optimization trials as a function of the number of model evaluations.

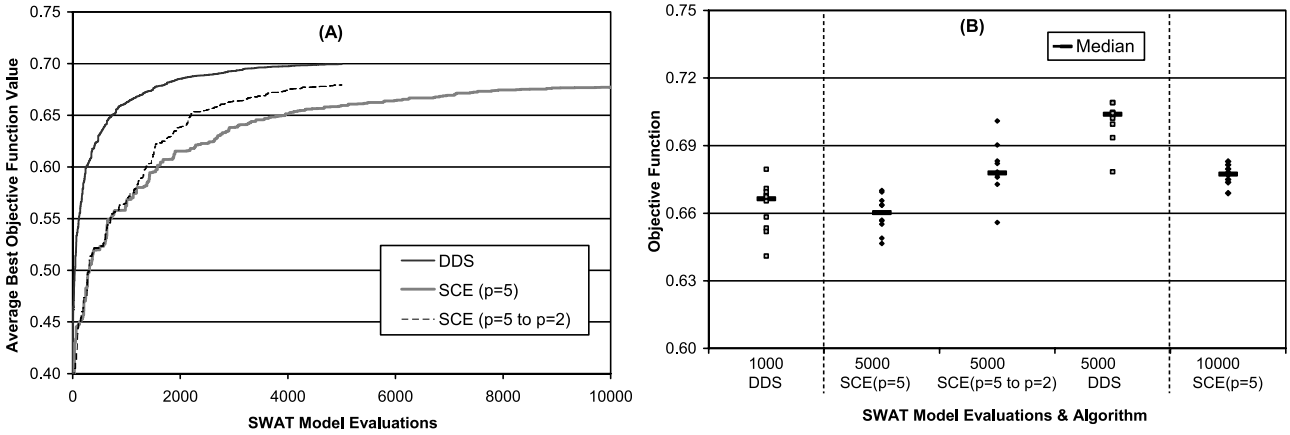

Figure 7. Algorithm performance comparison for the 26-D Town Brook simultaneous flow, sediment and phosphorus calibration maximization problem (formulation 4): (a) Average best objective function value (equation (3)) across 10 optimization trials as a function of the number of model evaluations. (b) All SCE and DDS optimization trial results after various numbers of model evaluations. *Sorooshian et al.* [1993] recommends SCE (p = 5 to p = 2) as efficient SCE strategy.

computationally effort (5000 DDS versus 10,000 SCE (p = 5) model evaluations), DDS was able to identify a solution that was slightly better than SCE for all three constituents. For formulation 5, using the same computational effort, the best DDS solution was substantially better than the best SCE solution in terms of the E_{NS} coefficient. For example, the daily sediment and total P E_{NS} coefficients for DDS were 0.20 and 0.10 higher, respectively, than the corresponding SCE values. In all cases in Table 4, the thresholds for the maximum absolute %Bias (30% and 20% for water quality constituents in formulation 4 and 5, respectively) were attained. In comparison with the DDS results from formulation 3 (calibration to Walton flows), where an average E_{NS} of 0.85 was obtained, the simultaneous inclusion of sediment and phosphorus in the objective function in formulation 5 only slightly reduced the flow E_{NS} coefficient to 0.82.

3.4. Sixpar Model Calibration Problem from *Duan et al.* [1993, 1992]

[53] The synthetic Sixpar six parameter hydrologic model calibration problem was an integral part of the original set of optimization problems used to introduce SCE [*Duan et al.*, 1993, 1992]. Although DDS was designed with larger dimensional problems in mind, DDS, along with our Matlab SCE, and the Fmincon algorithm were applied to the Sixpar problem available with the original SCE-UA Fortran code. SCE-UA results on the Sixpar problem are compared to the new results generated in this study. Our Matlab SCE handles the linear Sixpar constraints the same way as SCE-UA, while the DDS algorithm repeatedly samples new candidate solutions (steps 3 and 4 in Figure 1) until the linear constraints are satisfied. The Fmincon algorithm accounts for the linear constraints explicitly as they can be input to the algorithm. *Duan et al.* [1993] defined successful convergence to the global minimum as SSE values less than 10^{-3}. Thus each of the Fmincon optimization trials was restarted at new random initial solutions until a restart terminated with an SSE value less than 10^{-3}. All results for this problem were generated using 100 optimization trials. Further comparison details and results measuring multiple aspects of algorithm performance are summarized in Table 5.

[54] The most interesting result in Table 5 is that Fmincon is substantially more efficient and effective than all other algorithms. For example, after a maximum of three restarts, all 100 Fmincon optimization trials converged and stopped after an average of only 409 total Sixpar model evaluations to the SSE target value of 10^{-3} or better. In comparison, SCE-UA with p = 2 only achieved the target SSE value in 79/100 optimization trials and took an average of almost three times as many Sixpar evaluations (1104) as Fmincon to achieve the SSE target. SCE-UA with p = 6 shows higher reliability (99/100 achieve SSE target) but requires almost eight times as many Sixpar evaluations as Fmincon. The success of Fmincon on Sixpar is in contrast to the poor Fmincon performance on the SWAT2000 model calibration problems. Sixpar does not appear to be as difficult a global

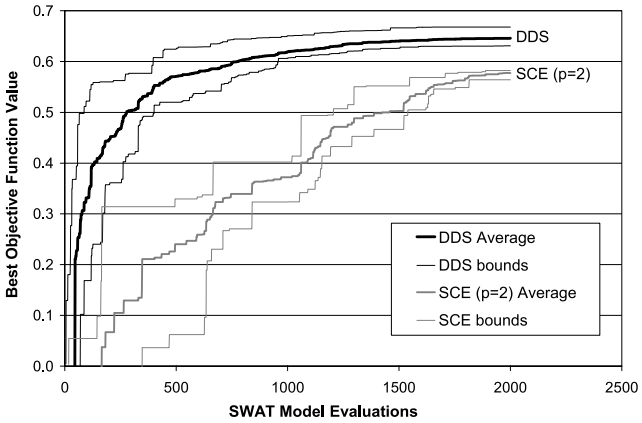

Figure 8. Algorithm performance comparison for the 30-D Walton/Beerston simultaneous flow, sediment and phosphorus calibration maximization problem (formulation 5): Average and bounds of best objective function value across five optimization trials as a function of the number of model evaluations. Bounds are based on the minimum and maximum best objective function value of the five optimization trials.

Table 4. Values of Weighted Objective Function Components (Flow, TSS, and Total P Performance Statistics) for the Best SCE and DDS Solutions to Formulations 4 and 5[a]

Formulation Number, Algorithm	Flow		TSS		Total P		Objective Function, E_w
	Daily Nash-Sutcliffe E_{NS}	Percent Bias[a]	Daily Nash-Sutcliffe E_{NS}	Percent Bias[a]	Daily Nash-Sutcliffe E_{NS}	Percent Bias[b]	
4, SCE (p = 5)[c]	0.65	−5.6%	0.72	−9.2%	0.71	−23.5%	0.68
4, DDS[c]	0.68	−5.5%	0.74	−4.0%	0.73	−22.8	0.71
5, SCE (p = 2)	0.79	−0.1%	0.34	4.9%	0.40	−12.8%	0.58
5, DDS	0.82	3.9%	0.54	16.7%	0.50	−8.0%	0.67

[a]See equations (3) and (4).
[b]All four solutions satisfied the maximum tolerable percent bias levels.
[c]DDS used 50% fewer model evaluations compared to SCE for formulation 4.

optimization test problem as the SWAT2000 calibration problems.

[55] The results in Table 5 do show that for this classical problem, SCE finds the global solution to the problem, much more effectively and efficiently than DDS. However, based on the equivalent E_{NS} calculated for the DDS and SCE solutions, the numerical difference between the DDS and SCE SSE values are not at all practically significant. For example, after 1104 Sixpar evaluations, DDS achieves an E_{NS} of 0.9996 in comparison with 1.0000 for SCE with p = 2. When flow measurement errors are considered in real calibration problems (where an E_{NS} of 0.8 is often considered to be good), such a small difference between the E_{NS} values is irrelevant. Note that our Matlab SCE shows an improved reliability over SCE-UA because unlike SCE-UA, it continues refining the solution after SCE-UA would have otherwise stopped due to additional SCE-UA convergence criteria.

4. Discussion

4.1. DDS and SCE Performance Comparison Summary

[56] Both SCE and DDS were developed as global optimizers to solve difficult watershed model calibration problems. No evidence has been presented to suggest that DDS is a better optimizer than SCE for lower-dimensional (six or fewer) problems. In addition, because comparisons here have been carried out largely from a limited total allowable function evaluation perspective, it is not clear which of DDS or SCE is a more appropriate optimizer for higher-dimensional problems when total allowable function evaluations are essentially unlimited (which is not typically the case for distributed watershed model calibration). Limited results in this study when 10,000 or more function evaluations were used provide examples where SCE performs better than DDS (e.g., 30-D Griewank and Ackley test functions) and where DDS performs better than SCE (e.g., Bump and 30-D Rastrigin test functions). In order to evaluate SCE and DDS suitability for automatic calibration of current computationally demanding watershed simulation models the remainder of the discussion will interpret results from a more limited total allowable function evaluation perspective.

[57] Available evidence in this study for the Sixpar model calibration example (Table 5) and the synthetic Town Brook flow calibration results in Figure 5 demonstrate numerically better SCE algorithm performance that was insignificant from a practical model calibration perspective. These two cases were the only model calibration problems in this study

Table 5. Comparative Algorithm Performance Averaged Over 100 Optimization Trials for the Sixpar Calibration Problem From *Duan et al.* [1992, 1993]

Algorithm[a]	Average SSE Across 100 Trials	E_{NS} Equivalent to Average SSE[b]	Number of Trials With SSE < 10^{-3}	AFE[c]	Average Sixpar Evaluations Across 100 Trials
Fmincon, 1 restart[d]	1.7689	0.9999	84	<314	399
Fmincon, 3 restarts[d]	0.0002	1.0000	100	< 409	409
SCE, p = 2	0.1857	1.0000	92	< 1104	1104[e]
SCE, p = 6	0.0002	1.0000	100	< 2118	2118
SCE-UA, p = 2[f]	NA	NA	79	1104	NA
SCE-UA, p = 6[f]	NA	NA	99	3133	NA
DDS[g]	5.3762	0.9996	0	NA	1104

[a]Here p is number of complexes.
[b]Calculated using the data in the Sixpar model example files as $E_{NS} = 1 - SSE_{avg}/12281$. A high value for E_{NS} is best. Maximum possible E_{NS} is 1.0000 (corresponding to global minimum of SSE = 0).
[c]Average Sixpar evaluations to reach SSE < 10^{-3}. Only trials achieving SSE < 10^{-3} included. The AFE performance metric was used by *Duan et al.* [1993].
[d]Fmincon restarts initialized to the best of 15 randomly selected parameter sets. The minimum difference for computing finite difference derivatives in Fmincon was set to 10^{-4} and the objective function and decision variable tolerances were set to 10^{-6} and 10^{-5}, respectively. Only three Fmincon trials needed to be restarted a third time. AFE is smaller than reported since Fmincon only stopped when convergence criterion met (often when SSE was much smaller than 10^{-3}) and intermediate results were not available.
[e]All 100 Matlab SCE optimization trials reached SSE < 10^{-3} in 1555 Sixpar evaluations or less.
[f]Available results taken from Table 2h of *Duan et al.* [1993]. NA means results not available.
[g]DDS results taken from optimization trials using 2500 Sixpar evaluations but results are only reported after 1104 Sixpar evaluations in order to compare against SCE, p = 2 for the same computational effort.

where SCE was observed to find better average final solutions than DDS. For the Sixpar results, very different conclusions about SCE versus DDS algorithm performance are reached when the strict numerical results (i.e., column 4 in Table 5) are used rather than the more practical calibration interpretation of results (i.e., column 3 in Table 5).

[58] In contrast, our results show that for all high-dimensional problems investigated here (those with 10 or more decision variables) the DDS algorithm offers quite substantial improvements over SCE performance for a wide range of total allowable objective function evaluations without requiring any algorithm parameter adjustment. DDS results were better than SCE in spite of the fact that the number of SCE complexes was fine-tuned and modified from default or recommended values for some problems in order to improve SCE performance (e.g., Figures 3, 4, 5, and 7). In 8 of 13 algorithm performance comparisons in this study, (including all four real SWAT2000 calibration formulations) the DDS algorithm produced better average solutions than SCE for the entire range of function or model evaluations considered (see Figures 2b, 3a, 4a, 5, 6, 7a, and 8). For SWAT2000 model calibration, the DDS performance advantage over SCE at various numbers of model evaluations was notable from a practical calibration perspective. For example, DDS bettered the average E_{NS} achieved by SCE by 0.05 to 0.15 for the range of model evaluations considered in Figure 6. In addition, in the 26 and 30 dimensional real calibration problems (see Figures 7a and 8), DDS required only 15–20% of the number of SCE model evaluations in order to find solutions with a better average objective function value than the final best average SCE objective function values.

[59] *Duan* [2003] recommends that SCE users experiment with the selection of algorithmic parameters on their own problem. Even if algorithmic parameter experimentation is limited to the number of complexes (p), conducting such experiments is clearly problematic when dealing with computationally expensive objective functions. *Madsen et al.* [2002] describe the problem of specifying the number of complexes for SCE when the number of model evaluations is limited as a general trade-off between efficiency and algorithm reliability and Figure 3 shows results from our case study highlighting this issue. No such algorithm parameter tuning is recommended when applying the DDS algorithm since DDS with a default r value of 0.2 was demonstrated to find relatively good solutions quickly for the range of dimensions and model (or objective function) evaluations considered in this study.

4.2. Summary of Previous SCE Algorithm Applications

[60] In the automatic watershed model calibration literature, the SCE algorithm has widely been considered for 13 years to be the standard for optimization as it is generally found to be robust, effective and efficient [*Duan*, 2003] and the original SCE publications [*Duan et al.*, 1993, 1992, 1994] are referenced in hundreds of publications. Further evidence of the stature of the SCE algorithm is the incorporation of SCE into advanced algorithms for uncertainty analysis [*Vrugt et al.*, 2003a, 2003b] and multiobjective optimization [*Vrugt et al.*, 2003a; *Yapo et al.*, 1998]. The contrast between this large body of SCE literature and our findings, which show that the new DDS algorithm outperforms the SCE algorithm for many problems, is somewhat surprising. It is therefore important to more closely review previous SCE studies in order to show that our SCE findings are not unreasonable or completely without precedent.

[61] In a recent review of global optimization for watershed models by *Duan* [2003], 20 studies were referenced that compare or apply the original SCE-UA algorithm for either watershed model calibration or application in other areas of hydrology. *Tolson* [2005] reviewed this set of SCE literature in order to look for similarities with the results reported in section 3 of this study. A subset of nine of these 20 studies [*Duan et al.*, 1993, 1992; *Franchini et al.*, 1998; *Gan and Biftu*, 1996; *Kuczera*, 1997; *Luce and Cundy*, 1994; *Sorooshian et al.*, 1993; *Tanakamaru and Burges*, 1996; *Thyer et al.*, 1999] compare SCE to other optimization algorithms. The majority of results in these nine publications only focus their comparison on final algorithm results after convergence and/or an extremely large number of model evaluations (often more than 10,000). Of the nine SCE comparison studies listed above, only two [*Kuczera*, 1997; *Tanakamaru and Burges*, 1996] present algorithm convergence rates versus the number of model evaluations similar to Figure 2. Presumably, this type of assessment led *Kuczera* [1997] to note GA performance was better than SCE in some limited cases. *Tolson* [2005] reports that results of *Tanakamaru and Burges* [1996] also show SCE to perform worse than at least one other algorithm if available model evaluations were limited to roughly 2000 or less.

[62] In summary, a closer analysis of some of the previous algorithm comparison studies involving SCE revealed that when total allowable objective function evaluations are limiting (i.e., SCE is not run to convergence), other algorithms have been found to perform equally well or better than SCE. For problem dimensions of ten or more, fewer than 10,000 function evaluations would generally be considered limiting based on the majority of SCE studies noted in section 1 and the paragraph above. There is a sizable suite of hydrologic or watershed simulation models, particularly distributed models, for which it is not practically feasible to conduct 10,000 or more model evaluations for calibration. The evidence in this paper combined with available evidence from earlier SCE comparative studies demonstrate that the ranking of SCE performance relative to other algorithms depends very much on the maximum number of model evaluations used for calibration.

4.3. DDS Algorithm Comments

[63] The default value of the neighborhood perturbation size parameter, r, of 0.2 produced good results across all the test functions and model calibration problems reported in section 3. These good results covered 6-, 10-, 14-, 20-, 26- and 30-dimensional optimization problems ranging mainly from 1000 to 10,000 total function evaluations. Therefore the default value for r seems robust and is suggested for most future DDS applications. In two of the relatively easier optimization problems considered in this study (the 10-D Griewank function because it has one main region of attraction and the synthetic SWAT2000 calibration problem (formulation 2) because error free measured data are used) reducing the r parameter from 0.2 to 0.1 improved DDS results. Another situation not investigated here that may call

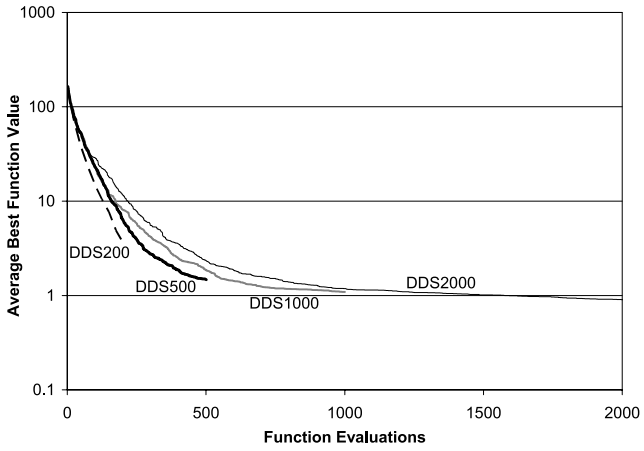

Figure 9. Average DDS behavior over 30 optimization trials as the maximum number of function evaluations (m) is varied for the 10-D Griewank function. DDS number means m set to that number of function evaluations. Same set of 30 initial solutions is used for each m.

for a decrease in the r parameter from 0.20 is when the default watershed model parameter set provides a reasonable objective function value such that modelers think it a good idea to concentrate the search closer to the default model parameter values. However, the probability DDS becomes mired near a poor local optima increases as r decreases.

[64] The DDS algorithm search strategy is scaled to the maximum number of total allowable function or model evaluations that the user wishes to expend on solving their problem. This scaling is designed to produce good DDS results after m function evaluations, regardless of whether m is 100 or 100,000. A demonstration of the results of this scaling behavior is shown in Figure 9 for the 10-dimensional Griewank function with the DDS neighborhood size parameter, r, set to the default of 0.2. Each set of DDS optimization trials in Figure 9 (DDS to 200, 500, 1000 and 2000 function evaluations) minimized the Griewank function 30 times from the same set of random initial solutions. Clearly, Figure 9 shows that the DDS algorithm effectively scales to any number of maximum function evaluations in a way that does not require users to determine unique r values for problems subject to different computational time constraints. The behavior of DDS depicted in Figure 9 was also observed for the test functions in Figure 2. From a calibration perspective, this DDS scaling behavior is attractive because it shows modelers do not necessarily need to experiment with DDS parameters if their total allowable model evaluations for calibration changes due to increased model complexity, time or space discretization, or more stringent modeling timelines.

5. Conclusions and Future Work

[65] For the range of test functions and watershed model calibration examples considered in this study, numerical results demonstrate that the DDS algorithm is a more computationally efficient and robust optimization algorithm than SCE in the context of distributed watershed model automatic calibration. DDS has been specifically shown to outperform SCE for multiple computationally intensive SWAT2000 model calibration examples. The value of DDS over SCE is greatest for computationally demanding models where the total number of model evaluations for calibration is limited and the number of calibrated parameters is high (10 or more). We have no conclusive evidence that DDS is better than SCE when 6 or fewer parameters are calibrated or an essentially unlimited number of model evaluations are used for calibration. In this study, the objective function or model evaluations were mainly limited to 10,000 or fewer and in 8 out of 13 optimization problems (including all four real SWAT2000 calibration formulations with 14 or more parameters calibrated), the DDS algorithm produced better average solutions than SCE for the entire range of function or model evaluations considered. A close review of previous SCE comparison literature from a distributed modeling perspective (i.e., total model evaluations for calibration are limited) show that our finding of relative SCE inefficiency is not without precedent.

[66] In the only automatic calibration formulation examples where SCE numerically outperformed DDS (two synthetic calibration examples) the numerical difference in terms of the common Nash-Sutcliffe calibration metric is too small to be of any interest (e.g., in the third or fourth decimal place). Furthermore, in the Sixpar synthetic calibration example [*Duan et al.*, 1992], a multistart derivative based method found the global minimum most reliably and efficiently. This shows that the Sixpar problem is not as difficult a global optimization problem as the SWAT2000 calibration problems introduced in this study.

[67] DDS is robust across a range of model calibration parameters (e.g., 6 to 30 in our examples) since it generated relatively good solutions without requiring any algorithm parameter adjustments. Results across multiple test functions show DDS can automatically scale to search for good calibration solutions within case study specific computational time limits without requiring algorithm parameter adjustment. The DDS algorithm is very simple and thus can be easily coded in whatever programming language is most convenient for the model being calibrated. Although this study focused on watershed models, the results are just as relevant to all environmental simulation modelers calibrating six or more parameters of a computationally demanding model.

[68] Algorithm comparisons presented here assessed DDS performance against SCE because SCE is currently the most commonly applied algorithm for automatic calibration of watershed simulation models. However, there have been substantial algorithmic advancements in the field of global optimization since SCE was introduced. It would be prudent to follow up this study with one that compares SCE, DDS, additional global optimization algorithms, and other efficiency-oriented optimization algorithms such as the environmental simulation model parameter estimation (PEST) method [e.g., *Doherty and Johnston*, 2003]. In addition, the authors are currently investigating and testing modifications to the DDS methodology to (1) improve algorithm performance on lower-dimensional (i.e., <10) optimization problems, (2) increase ability of DDS to locate the exact global optimum, and (3) implement a parallelized version of DDS. The authors have also incorporated DDS

into a new and efficient approximate uncertainty analysis methodology [*Tolson*, 2005].

[69] Matlab and Fortran 90 source codes for DDS, as well as a compiled DDS program linkable via text files to user-specific objective functions, are available by emailing the first author.

[70] **Acknowledgments.** This research was a part of B. Tolson's Ph.D. research at Cornell University that was supported in part by NSF grant BES-0229176 to C. Shoemaker. The Cannonsville watershed modeling example was derived from an updated version of a model described in an earlier report that was supported by Safe Drinking Water EPA funds to NY Department of Environmental Conservation and DCAP in Delaware County. The authors would like to thank Fernando Méndez for his work coding the SCE algorithm in Matlab, Qingyun Duan for providing the SCE-UA and Sixpar model source code in Fortran, and both Jery Stedinger and Pradeep Mugunthan for providing insightful comments on the work summarized in this manuscript. Last, the authors appreciate the constructive comments from three reviewers and associate editor that improved the presentation of these findings.

References

Ackley, D. H. (1987), *A Connectionist Machine for Genetic Hillclimbing*, Springer, New York.

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

Arnold, J. G., R. Srinivasan, R. R. Muttiah, and J. R. Williams (1998), Large area hydrologic modeling and assessment part I: Model development, *J. Am. Water Resour. Assoc.*, *34*, 73–89.

Doherty, J., and J. M. Johnston (2003), Methodologies for calibration and predictive analysis of a watershed model, *J. Am. Water Resour. Assoc.*, *39*, 251–265.

Duan, Q. (2003), Global optimization for watershed model calibration, in *Calibration of Watershed Models, Water Sci. Appl.*, vol. 6, edited by Q. Duan et al., pp. 89–104, AGU, Washington, D. C.

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

Duan, Q., V. K. Gupta, and S. Sorooshian (1993), Shuffled complex evolution approach for effective and efficient global minimization, *J. Optim. Theory Appl.*, *76*, 501–521.

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

Franchini, M., G. Galeati, and S. Berra (1998), Global optimization techniques for the calibration of conceptual rainfall-runoff models, *Hydrol. Sci. J.*, *43*, 443–458.

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

Griewank, A. O. (1981), Generalized descent for global optimization, *J. Optim. Theory Appl.*, *34*, 11–39.

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

Ibbitt, R. P. (1970), Systematic parameter fitting for conceptual models of catchment hydrology, Ph.D. thesis, Univ. of London, U. K.

Keane, A. J. (1996), A brief comparison of some evolutionary optimization methods, in *Modern Heuristic Search Methods*, edited by V. J. Rayward-Smith et al., pp. 255–263, John Wiley, Hoboken, N. J.

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

Luce, C. H., and T. W. Cundy (1994), Parameter-identification for a runoff model for forest roads, *Water Resour. Res.*, *30*, 1057–1069.

Madsen, H., G. Wilson, and H. C. Ammentrop (2002), Comparison of different automated strategies for calibration of rainfall-runoff models, *J. Hydrol.*, *261*, 48–59.

Masri, S. F., G. A. Bekey, and F. B. Safford (1980), A global optimization algorithm using adaptive random search, *Appl. Math. Comput.*, *7*, 353–375.

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

Neitsch, S. L., J. G. Arnold, J. R. Kiniry, and J. R. Williams (2001), Soil and Water Assessment Tool user's manual version 2000, report, U.S. Dept. of Agric. Agric. Res. Serv., Temple, Tex.

Nelder, J. A., and R. Mead (1965), A simplex method for function minimization, *Comput. J.*, *7*, 308–313.

Price, W. L. (1978), A controlled random search procedure for global optimization, in *Towards Global Optimization 2*, edited by C. W. L. Dixon and G. P. Szego, pp. 71–84, Elsevier, New York.

Rastrigin, L. A. (1974), *Systems of Extremal Control* (in Russian), Nauka, Moscow.

Schwefel, H.-P. (1995), *Evolution and Optimum Seeking*, John Wiley, Hoboken, N. J.

Singh, V. P., and D. A. Woolhiser (2002), Mathematical modeling of watershed hydrology, *J. Hydrol. Eng.*, *7*, 270–292.

Sorooshian, S., and V. K. Gupta (1983), Automatic calibration of conceptual rainfall-runoff models: The question of parameter observability and uniqueness, *Water Resour. Res.*, *19*, 251–259.

Sorooshian, S., Q. Y. Duan, and V. K. Gupta (1993), Calibration of rainfall-runoff models—Application of global optimization to the Sacramento Soil-Moisture Accounting Model, *Water Resour. Res.*, *29*, 1185–1194.

Tanakamaru, H., and S. J. Burges (1996), Application of global optimization to parameter estimation of the tank model, paper presented at the International Conference on Water Resources and Environment Research, Water Resour. Cent., Kyoto Univ., Kyoto, Japan, 29–31 Oct.

Thyer, M., G. Kuczera, and B. C. Bates (1999), Probabilistic optimization for conceptual rainfall-runoff models: A comparison of the shuffled complex evolution and simulated annealing algorithms, *Water Resour. Res.*, *35*, 767–773.

Tolson, B. A. (2005), Automatic calibration, management and uncertainty analysis: Phosphorus transport in the Cannonsville watershed, Ph.D. thesis, Cornell Univ., Ithaca, N. Y.

Tolson, B. A., and C. A. Shoemaker (2004), Watershed modeling of the Cannonsville Basin using SWAT2000: Model development, calibration and validation for the prediction of flow, sediment and phosphorus transport to the Cannonsville reservoir, version 1.0, technical report, Sch. of Civ. and Environ. Eng. Cornell Univ., Ithaca, N. Y. (Available at http://techreports.library.cornell.edu:8081/Dienst/UI/1.0/Display/cul.wat/2004-2)

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

Vrugt, J. A., H. V. Gupta, L. A. Bastidas, W. Bouten, and S. Sorooshian (2003a), Effective and efficient algorithm for multiobjective optimization of hydrologic models, *Water Resour. Res.*, *39*(8), 1214, doi:10.1029/2002WR001746.

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

Wang, Q. J. (1991), The genetic algorithm and its application to calibrating rainfall-runoff models, *Water Resour. Res.*, *27*, 2467–2471.

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

C. A. Shoemaker, School of Civil and Environmental Engineering, Cornell University, 210 Hollister Hall, Ithaca, NY 14853, USA. (cas12@cornell.edu)

B. A. Tolson, Department of Civil and Environmental Engineering, University of Waterloo, 200 University Avenue West, Waterloo, ON, Canada N2L 3G1. (btolson@uwaterloo.ca)