Multiple-Objective Stepwise Calibration Using Luca

By Lauren E. Hay and Makiko Umemoto

Open-File Report 2006-1323

U.S. Department of the Interior
U.S. Geological Survey

U.S. Department of the Interior
DIRK KEMPTHORNE, Secretary

U.S. Geological Survey
Mark D. Myers, Director

U.S. Geological Survey, Reston, Virginia 2006
Revised and published: 2007

For product and ordering information:
World Wide Web: http://www.usgs.gov/pubprod
Telephone: 1-888-ASK-USGS

For more information on the USGS—the Federal source for science about the Earth,
its natural and living resources, natural hazards, and the environment:
World Wide Web: http://www.usgs.gov
Telephone: 1-888-ASK-USGS

Suggested citation:
Hay, L.E., and Umemoto, Makiko, 2007, Multiple-objective stepwise calibration using Luca: U.S. Geological Survey Open-File Report 2006-1323, 25 p.

CONTENTS

FIGURES

Tables

Multiple-Objective Stepwise Calibration Using Luca

By Lauren E. Hay and Makiko Umemoto

Abstract

This report documents Luca (**L**et **u**s **ca**librate), a multiple-objective, stepwise, automated procedure for hydrologic model calibration and the associated graphical user interface (GUI). Luca is a wizard-style user-friendly GUI that provides an easy systematic way of building and executing a calibration procedure. The calibration procedure uses the Shuffled Complex Evolution global search algorithm to calibrate any model compiled with the U.S. Geological Survey's Modular Modeling System. This process assures that intermediate and final states of the model are simulated consistently with measured values.

Introduction

Is this report, Luca (**L**et **u**s **ca**librate), a multiple-objective calibration strategy that incorporates additional data sets, is presented. This approach uses the Shuffled Complex Evolution (SCE) global search algorithm (Duan and others, 1994) to calibrate parameters for any model compiled within the Modular Modeling System (MMS) (Leavesley and others, 1996). MMS is a set of modular modeling tools used to address the problems of model selection, application, and analysis. When using MMS, the user can selectively couple the most appropriate process algorithms from applicable models to create an "optimal" model for the desired application. Intermediate variables (such as solar radiation, potential evapotranspiration, snow-water equivalent, snow-covered area, and soil moisture) and final variables (such as the water balance and components of the daily hydrograph) simulated by the selected MMS model can be considered a calibration data set if there is an associated "observed" variable that can be used for calibration. The parameters influencing each of the selected model variables are calibrated in a multiple-objective, stepwise procedure.

The use of multiple-objective functions in the calibration of hydrologic models has become increasingly popular. For example, Hogue and others (2000) examined recessions and low flows, higher flows, and base flows; Turcotte and others (2000) examined droughts, annual and monthly flow volumes, high flows, high-flow synchronization, and snowmelt runoff; Madsen (2000) examined the water balance, hydrograph shape, peak flows, and low flows; and Boyle and others (2000; 2003) examined three components of the hydrograph described as driven, non-driven-quick, and non-driven-slow. While these studies used multiple objectives, the only variable used for calibration was derived from runoff, the final model state. Different portions and time-steps of the hydrograph were configured for these multiple-objective calibrations. This traditional approach to calibration and evaluation of distributed-hydrologic models—comparison of observed and simulated runoff—is not sufficient by itself in model evaluation (Refsgaard, 1997). Intermediate

variables computed by the hydrologic model could be characterized by parameter values that do not replicate those hydrological processes in the physical system.

Luca facilitates the calibration of intermediate and final variables produced by the selected MMS model, giving the user higher confidence in the model output by assuring that the chosen intermediate and final states of the model are simulated consistently with "observed" values. Luca has been used successfully in a number of recent applications. In Hay and others (2006a), the calibration procedure was introduced, including the sequential calibration of a distributed hydrologic model's simulation of solar radiation (SR), potential evapotranspiration (PET), water balance, and daily runoff. The procedure used the Shuffled Complex Evolution (SCE) global search algorithm (Duan and others, 1992; 1993; and 1994) to calibrate the U.S. Geological Survey's Precipitation Runoff Modeling System (Leavesley and others, 1983) in the Yampa River basin of northwestern Colorado. This process assured that intermediate states of the model (SR and PET on a monthly mean basis), as well as the water balance and components of the daily hydrograph are simulated consistently with "observed" values. Hay and others (2006c) used the same sequential calibration for hydrologic model calibration, implementing a one-way coupling of an atmospheric and a hydrologic model in Colorado.

Hay and others (2006b) applied the multiple-objective, stepwise, automated procedure to a mountainous watershed in the Upper Klamath Basin of Oregon. The procedure includes the sequential calibration of simulated (1) SR, (2) PET, (3) annual water balance; (4) snow-covered area; and (5) components of daily runoff. The multiple-step calibration procedure ensured that intermediate model states and fluxes, as well as the annual water balance, components of the daily hydrograph, and snow-covered area were being simulated consistent with "observed" values. In comparison to models calibrated by using streamflow data alone, this sequential calibration procedure was found to produce model parameter sets that are more reliable for hydrologic data assimilation.

As evidenced by the above studies, numerous calibrations were required when using the same multiple-step setup. In the following sections, detailed instructions of Luca, including all available functionalities, are presented.

Luca

Luca's wizard-style user-friendly graphical user interface (GUI) is written in Java and provides an easy systematic way of building a multiple-objective, stepwise calibration procedure. Luca uses the Shuffled Complex Evolution (SCE) global search algorithm (Duan and others, 1994) to calibrate parameters for any model compiled within the Modular Modeling System (MMS) (Leavesley and others, 1996). To run Luca, Java and MMS must be installed. For information on installing MMS, the user is referred to the MMS user's manual (Leavesley and others, 1996).

To invoke Luca, one of the following commands is used:

1. For cygwin: java –cp `cygpath –wp oui.jar:jhall.jar` oui.sce.gui.SCEWizardFrame
2. For UNIX/LINUX: java -classpath oui.jar:jhall.jar oui.sce.gui.SCEWizardFrame

After invoking the appropriate command, the window in figure 1 with the first set of instructions should appear. If a Windows platform is used, cygwin will need to be installed (*http://www.cygwin.com*).

Each window in Luca consists of three components (see fig. 1):
(1) an instruction panel on (the orange colored panel located on the left in fig. 1)
(2) a main panel (outlined with a magenta box in fig. 1)
(3) five buttons (outlined with a green box in fig. 1)

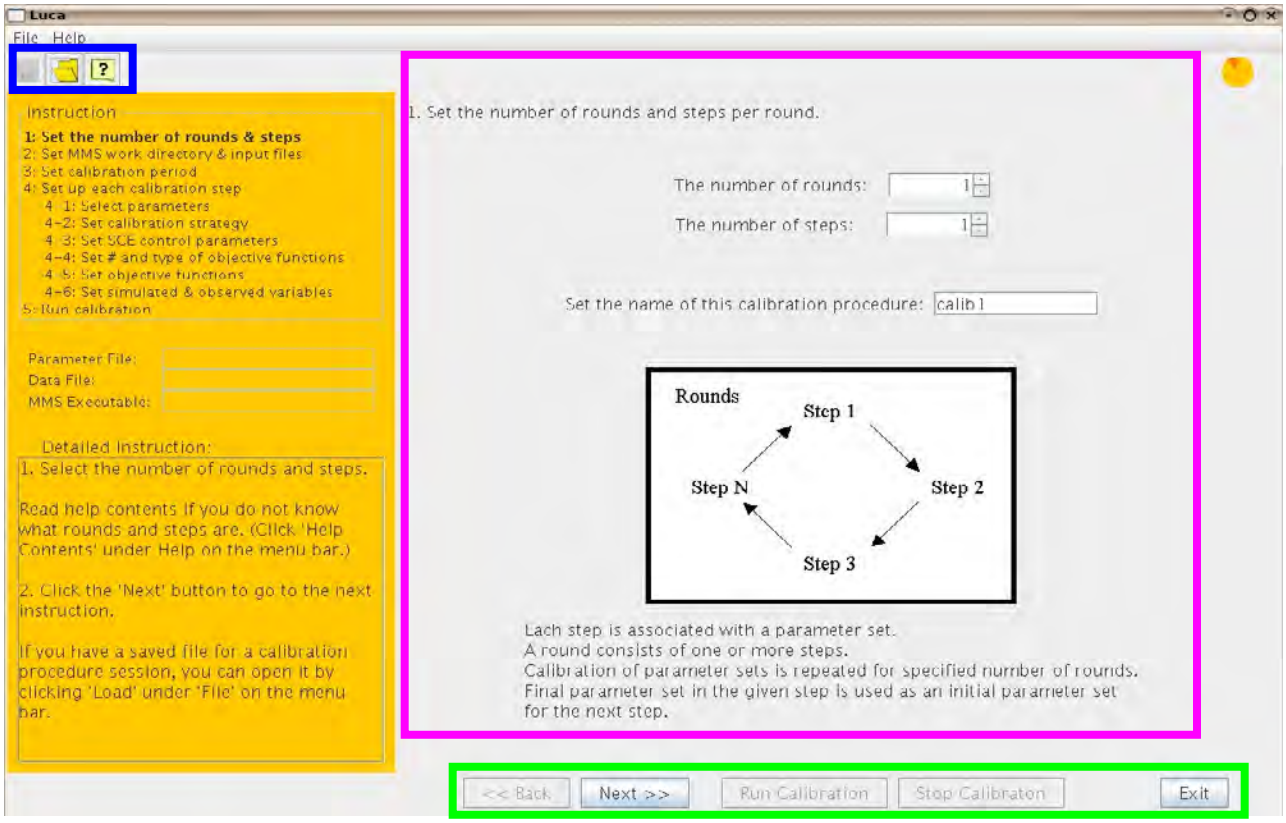

Figure 1. Instruction 1 in Luca.

The orange instruction panel provides an overview of all tasks the user has to complete in order to build the calibration procedure. This panel also shows the current instruction indicated by the bold letters (the current instruction in Fig. 1 is Instruction 1: Set the number of rounds & steps). User chosen names will appear in the **Parameter file**, **Data File**, and **MMS executable** boxes. Specific details (**Detailed Instruction**) for the current instruction appear at the bottom of this panel.

The user builds a calibration procedure by following the instructions in the main panel (outlined with a magenta box in fig. 1). Default values will appear in the main panel when available. The user can choose to use defaults or replace them with user-defined values.

Five buttons are located at the bottom of every Luca window (outlined with a green box in fig. 1). A button is highlighted if active in the current window. The user can proceed to the next instruction by clicking the **Next** button. The **Back** button allows the user to go back to a previous instruction. The **Exit** button allows the user to exit from Luca. The **Run Calibration** and **Stop Calibration** buttons are described in **Instruction 5**.

Luca provides three additional functionalities: save, load, and help. These are located in the blue box in figure 1. After the calibration procedure is built in Instruction 5, the session can be saved by clicking the Save button (computer disk schematic; left button) and specifying the filename in which to save the session. This session file can be loaded at any time by clicking the Load button (open file schematic; middle button) and specifying the session file to load. When the Help button (? Schematic; right button) is clicked, the user has access to the Luca help files. These same functionalities are available through the “File” and “Help” buttons in the upper left-hand

corner. The “File” button contains the save, load, and font functionalities. The save and load functionalities are similar to those just described. The text font option allows the user to change the font of the GUI. The “Help” button gives the user access to the Luca help files.

Table 1. Overview of Luca.

[MMS, Modular Modeling System; SCE, Shuffled Complex Evolution]

Instruction Number	Detailed Instructions	
Instruction 1 *Set the number of rounds and steps*	Set the number of rounds, number of steps per round, and the name of the calibration procedure	
Instruction 2 *Set MMS work directory and input files*	Select the MMS work directory (mms_work), input-parameter file, input-data file, and executable.	
Instruction 3 *Set calibration period*	Set the start and end date of calibration period (first year is used for model initialization).	
Instruction 4 *Set up each calibration step*	Instruction 4-1 *Select parameters*	Choose parameters to calibrate from list of available parameters for each step
	Instruction 4-2 *Set calibration strategy*	For each parameter selected for each step in the calibration indicate whether: (1) to include the initial point in the population; (2) the parameter should be treated as a mean, individual, or binary treatment; (3) the lower and upper bound for the calibration; and (4) optional file name for calibration parameter values.
	Instruction 4-3 *Set SCE control parameters*	The following SCE control parameters are set: (1) Number of complexes in the initial population; (2) number of points in each complex; (3) number of points in a sub-complex; (4) number of evolution steps before shuffling; (5) minimum number of complexes; (6) maximum number of model executions; (7) number of shuffling loops in which the criterion value must change by a given percent before optimization is terminated; and (8) given percentage for the criterion value. The “Restore Defaults” option can be clicked for default control parameter values if desired. The “Let me modify values” button must be clicked if the user is running on Linux.
	Instruction 4-4 *Set number and type of objective functions*	For each step, select the number and type of objective functions.
	Instruction 4-5 Set objective functions	For each step, set the weights and the time step associated with each objective function. The “data-subdivide” option can be selected to allow for calibration of specific time periods within the designated calibration period.
	Instruction 4-6 *Set simulated and observed variables*	Select the simulated and observed variables for each step. If the observed data is from an external source, then select whether it occurs as a single value or a range for each time step.
Instruction 5 *Run calibration*	Select the “Run Calibration” button to start the calibration process.	

The following section describes Luca in detail. Table 1 gives a general description of the five instructions for building a calibration procedure by using Luca. Each of the following sections describes in detail the instructions listed in table 1.

Luca Instruction 1: Set the number of rounds and steps per round

Luca requires a user-defined number of steps, which are executed sequentially within a user-defined number of rounds. To start the calibration procedure, an initial parameter file containing all MMS parameters is defined. The parameters identified for each calibration step are calibrated. These calibrated parameter values replace the respective parameter values in the parameter file, and this parameter file is used as the initial parameter file for the next calibration step. Completion of the user-designated number of steps constitutes a round. Once a parameter is calibrated, its value is set for the remainder of that calibration round. This process is repeated until the user-designated number of steps and rounds are completed.

Instruction 1 in Luca, **Set the number of rounds & steps,** is shown in figure 1. The user is prompted to set **The number of rounds** and **The number of steps**. A step is associated with a selection of parameters from a given input-parameter file. A round consists of one or more steps (see figure insert in fig. 1).

A name for the calibration procedure is placed in the **Set the name of this calibration procedure** field. Assigning a unique calibration-procedure name allows the user to run several calibration procedures simultaneously. The name placed in this field will be added to all of Luca's output-file names.

Table 2 shows the Luca output parameter files produced from a calibration procedure named "DAISY" with "M" rounds and "N" steps. In Step 1 of Round 1, a user-defined, input parameter file is used, and the parameter values selected for Step 1 are calibrated. The calibrated values are written to an output parameter file. In Step 2 of Round 1, the output parameter file produced in Step 1 of Round 1 is used as the input parameter file for Step 2. The selected parameters for Step 2 are calibrated, and an output parameter file is produced. The same procedure is repeated for Step 3, 4, …., "N" of Round 1. In Step 1 of Round 2, the output parameter file produced in Step "N" of Round 1 is used as the input parameter file for Step 1 of Round 2. The selected parameters for Step 1 are calibrated, and an output parameter file is produced. This procedure is repeated for the remaining steps and rounds.

At the end of the calibration procedure, the output parameter file from the last step ("N") of the last round ("M") is the final parameter file (roundM_DAISY_stepN.par in table 2). Luca creates a summary file (DAISY_summary.txt in table 2) with information on each step and round of the calibration procedure. An MMS statvar file (DAISY_luca.statvar in table 2) contains the final time-series output for the calibration procedure using the final parameter set (roundM_DAISY_stepN.par in table 2). The statvar file is an MMS output file as defined in the MMS user's manual (see Leavesley and others, 1996).

Table 2. Files produced from Luca procedure named "DAISY" with "M" rounds and "N" steps.
[Summary file name for calibration procedure: DAISY_summary.txt; Modular Modeling System statvar output file with timeseries data: DAISY_luca.statvar]

Round Number	Step Number	Input parameter file name	Output parameter file name
1	1	User defined	round1_DAISY_step1.par
	2	round1_DAISY_step1.par	round1_DAISY_step2.par
	N	round1_DAISY_step2.par	round1_DAISY_stepN.par
2	1	round1_DAISY_stepN.par	round2_DAISY_step1.par
	2	round2_DAISY_step1.par	round2_DAISY_step2.par
	N	round2_DAISY_step2.par	round2_DAISY_stepN.par
M	1	round2_DAISY_stepN.par	roundM_DAISY_step1.par
	2	roundM_DAISY_step1.par	roundM_DAISY_step2.par
	N	roundM_DAISY_step2.par	roundM_DAISY_stepN.par

Luca Instruction 2: Select the Modular Modeling System work directory, input parameter file, data file, and executable

Instruction 2 in Luca, **Set MMS work directory & input files,** is shown in figure 2. The user selects the **MMS Work Directory** (mms_work) by using the **Browse** button. The mms_work directory must be defined according to the instructions outlined in Leavesley and others (1996). When the mms_work directory is selected in Instruction 2, the contents in the mms_work/input/params, mms_work/input/data, and mms_work/models directories are displayed in the Luca screen (see fig. 3). A **Parameter File**, **Data File**, and **MMS Model Executable** must be selected from the mms_work/input/params, mms_work/input/data, and mms_work/models directories, respectively.

When the parameter, data, and model-executable files are chosen, the file names will appear throughout the rest of the calibration procedure in the orange box on the left in figure 3. In figure 3, the parameter file is PRMS_parameters, the data file is PRMS_input.data, and the MMS executable is xprms.

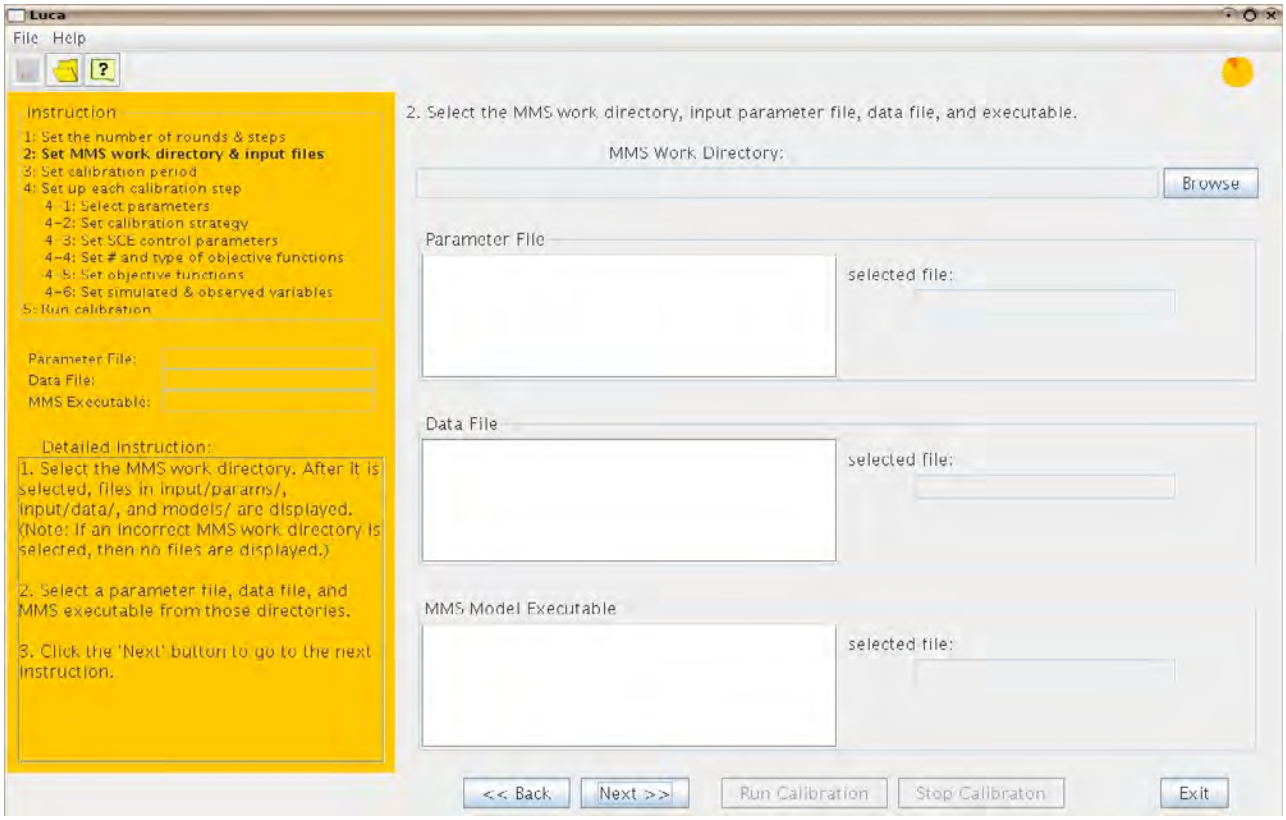

Figure 2. Instruction 2 in Luca.

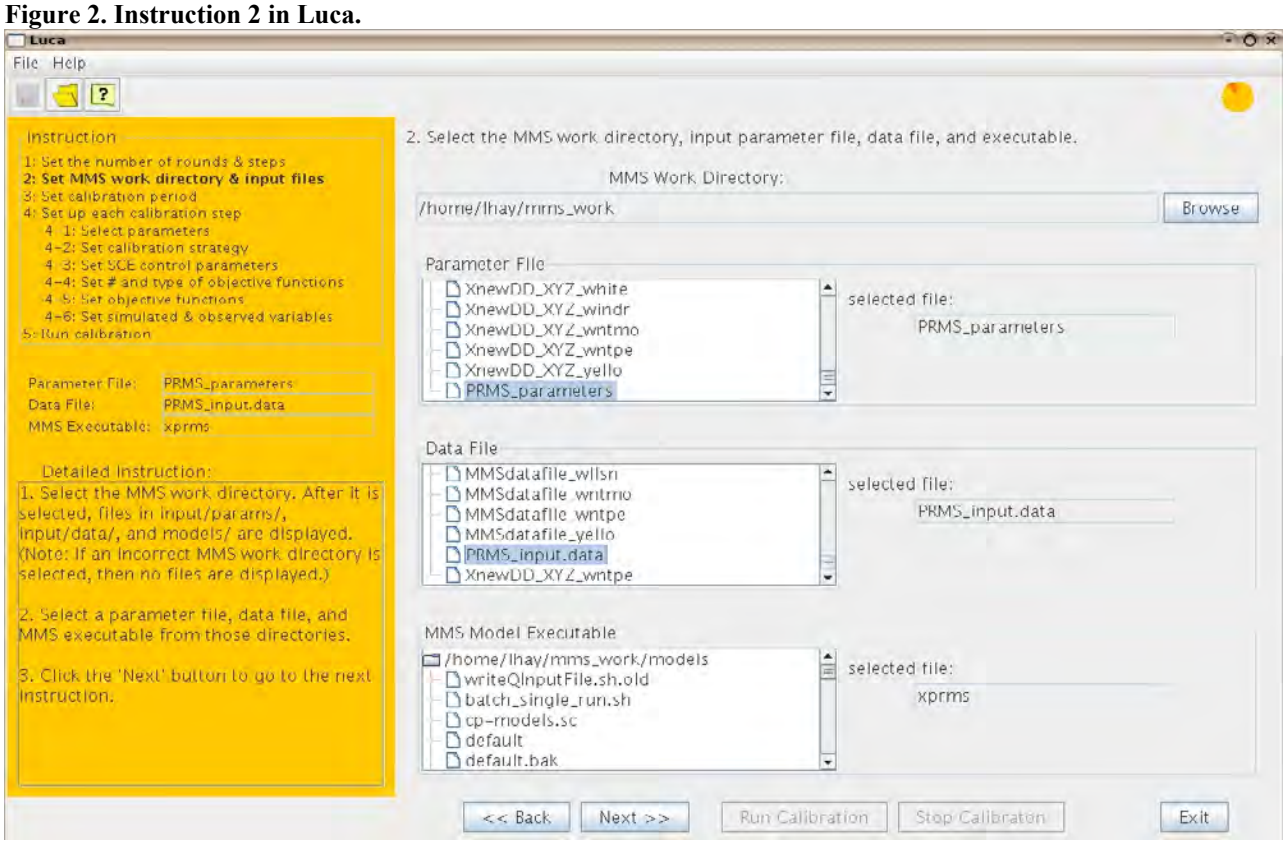

Figure 3. Modular Modeling System (MMS) work directory, parameter file, data file, and MMS model executable selection.

Luca Instruction 3: Set start and end date of calibration period

Instruction 3 in Luca, **Set calibration period,** is shown in figure 4. The user sets the **Start Date** and **End Date** of the calibration period in this instruction. Luca examines the MMS input data file designated in Instruction 2 and lists the corresponding start and end dates when Instruction 3 is initially displayed. The user can modify the number of years used for the calibration period.

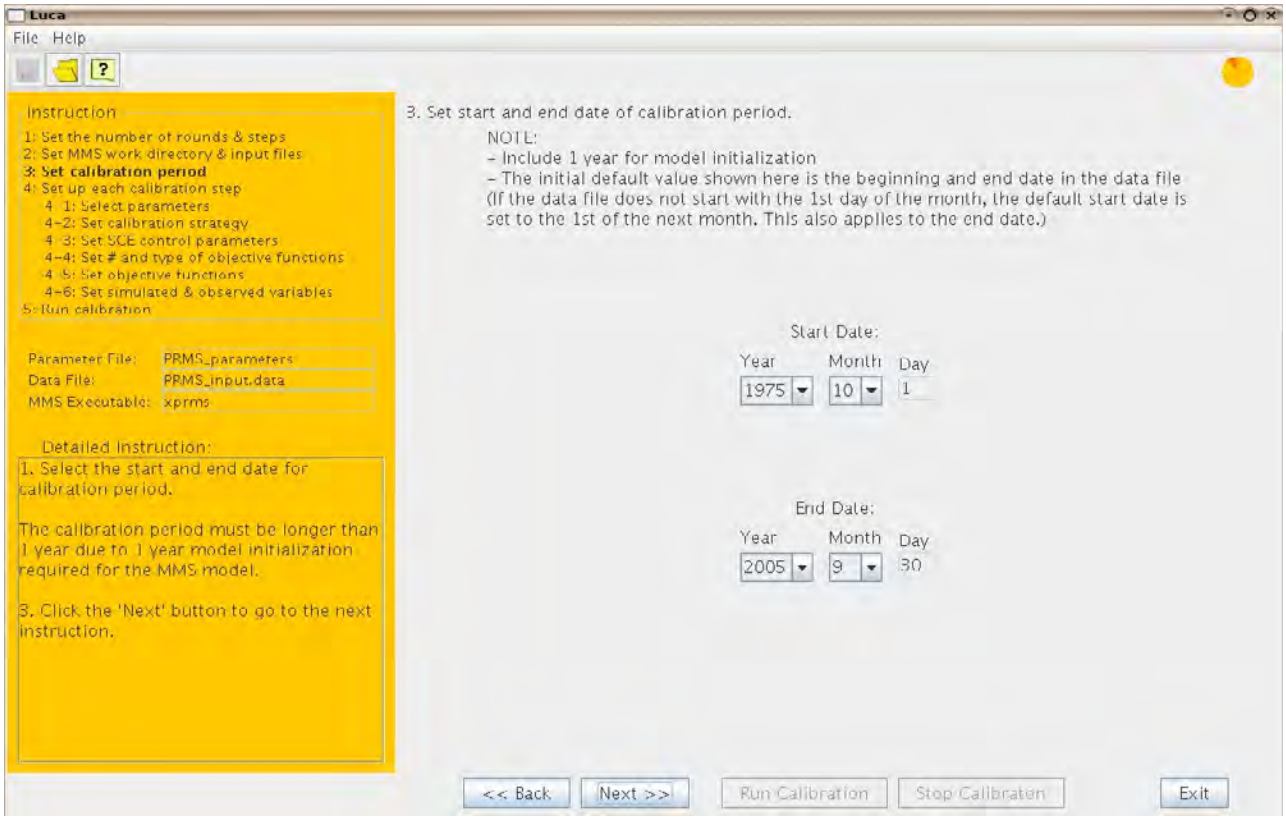

Figure 4. Instruction 3 in Luca.

The calibration period is the period for which the MMS model performs a simulation. One year for model initialization is included in the specified calibration period. This means that the MMS model runs for the given calibration period; however, the first year of the values are not included in the objective function calculation. For example, in figure 4 the calibration period is set to 1975/10/1 – 2005/9/30. This means the MMS executable will run for the period 1975/10/1 – 2005/9/30 and the period 1976/10/1 – 2005/9/30 will be used for the objective-function calculations.

Luca Instruction 4: Set up each calibration step

Instruction 4 in Luca, **Set up each calibration step,** has six sets of instructions (4-1 to 4-6). In each of these six instructions, tabs are displayed on top of the main panel, indicated by the magenta arrow in figure 5. Each tab is associated with a step. The single tab shown on the top of figure 5 (**Step 1**) indicates that there is one step (set in Instruction 1, fig. 1.) Luca allows the user to establish different settings for each step by selecting the appropriate step tab and completing the identical instructions under each step tab. The tab for the current step being edited will appear highlighted.

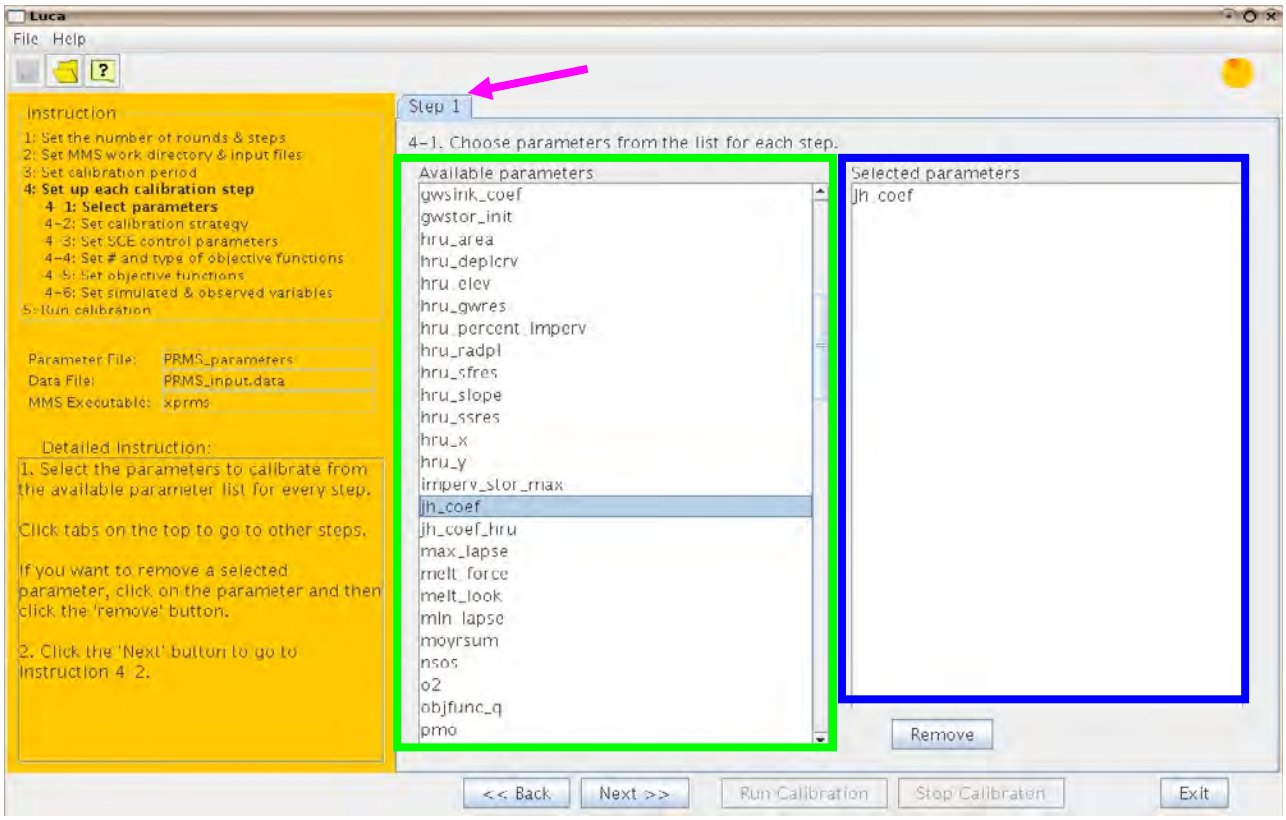

Figure 5. Instruction 4-1 in Luca.

Instruction 4-1: Choose parameter from the list for each step

Instruction 4-1 in Luca, **Select parameters**, is shown in figure 5. For each step, the user must **Choose parameters from the list for each step.** The list of **Available parameters** (green box in the main panel in fig. 5) is determined from the MMS input parameter file selected in Instruction 2. The **Selected parameters** are shown on the right in the main panel (blue box in fig. 5). **Selected parameters** can be highlighted and removed as necessary by using the **Remove** button. A distinct calibration-parameter set is made for each step by clicking on the step tabs and completing the identical instructions for each step.

The parameters selected in each step need to have an effect on the variable chosen as the calibration data set. This can be determined through a parameter-sensitivity analysis. Parameters

are not included in this step that do not have any effect on the simulated variable (chosen in Instruction 4-6).

Instruction 4-2: Select the calibration strategy for each parameter

Instruction 4-2 in Luca, **Set the calibration strategy**, is shown in figure 6. Similar to Instruction 4-1, the user must click on the step tabs and complete Instruction 4-2 for each step. If the user changes any of the initial parameter values, the original parameter file can be saved by inserting a name in the green box in figure 6. Note that Luca does not use this file. Luca keeps parameter files for all steps and rounds (see table 2 for default names). The output parameter files will be placed in the user's mms_work/input/params directory. A final MMS output file is not saved for each step of each round. The MMS model selected in Luca keeps writing the same output file ("DAISY_luca.statvar" in table 2) while running. When necessary, a final MMS output file for each step of each round can be reproduced by using the parameter files created by Luca.

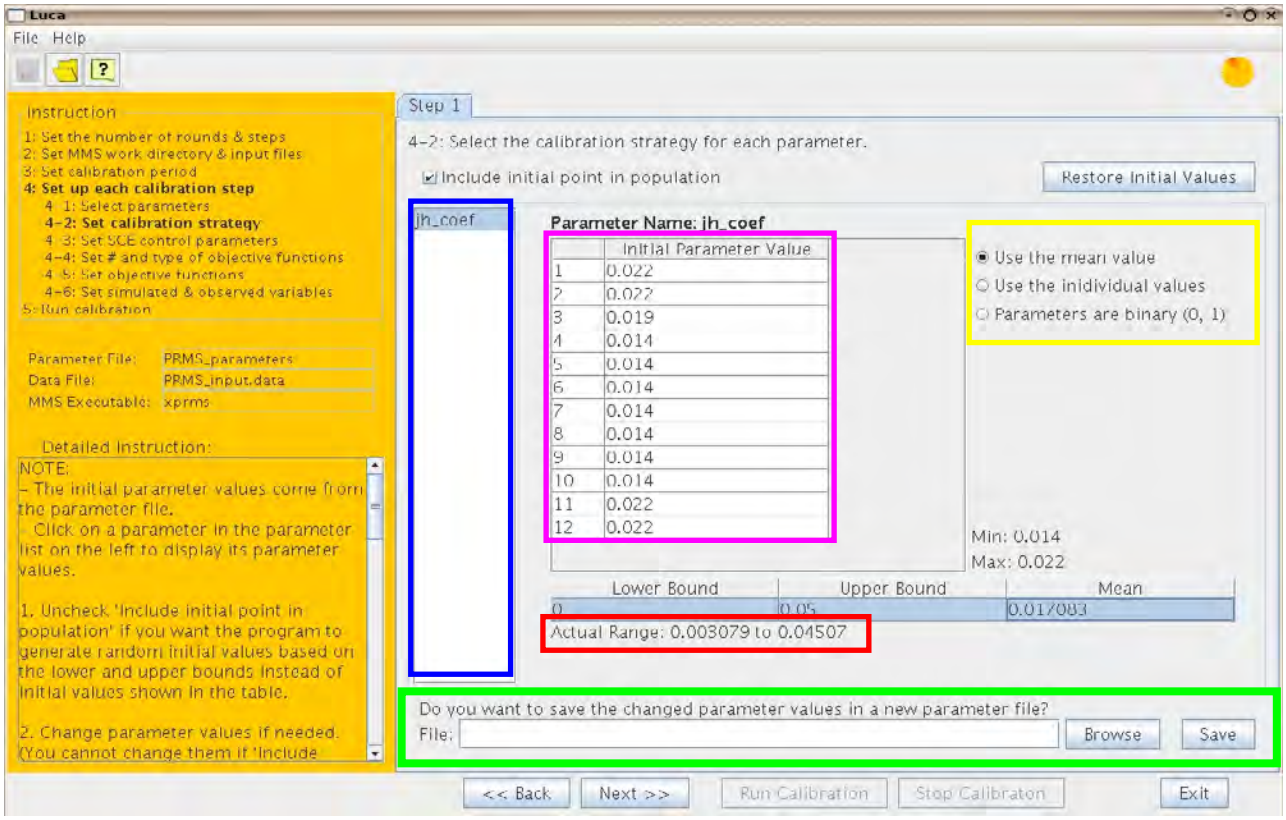

Figure 6. Instruction 4-2 in Luca. "Use the mean value" is chosen.

The check box, **Include initial point in population,** needs to be checked if the user wants the initial parameter values in that step to be used in the global search algorithm (SCE, described in Instruction 4-3). If this box is left unchecked, then all points are generated by SCE for that step.

The panel on the left side of the main panel in figure 6 (blue box) shows the parameters selected for calibration (in Instruction 4-1). In this case, one parameter, jh_coef, was selected in Instruction 4-1. When the parameter name is highlighted, the **Initial Parameter Values** are shown in the center panel (magenta box). These initial parameter values are determined from the MMS

input parameter file selected in Instruction 2. The user has an option of changing the initial values by clicking on the values and typing the new value.

On the right in figure 6 (yellow box), the user has three options for each parameter chosen in Instruction 4-1: **Use the mean value**; **Use the individual values**; or **Parameters are binary (0,1).**

When **Use the mean** is chosen, the mean parameter value (instead of each individual parameter value) is calibrated. Each time SCE generates a value for the mean, individual parameter values are generated based on the new mean such that the mean-value distribution is preserved. This option is a good choice when a spatially distributed parameter is chosen for calibration.

When the mean value is chosen as the calibration type, the individual parameter values must be regenerated from the SCE-generated mean value. Given n-individual initial parameter values ($Pinit_n$), the new individual parameter values (P_n) are reproduced from the SCE-generated mean ($MEAN_{SCE}$) by using the following equation:

$$P_n = \{[(MEAN_{SCE} + C) * (Pinit_n + C)\ ] / [MEAN_{INIT} + C]\} - C \qquad (1)$$

where $MEAN_{INIT}$ is the mean of n-initial parameters ($Pinit_n$). C is a constant used to avoid zero values in equation (1):

$$C = [\text{absolute value of the user defined lower bound}] + 10. \qquad (2)$$

In figure 6, **Use the mean value** has been chosen, the jh_coef parameter is highlighted, and the initial parameter values are shown (12 monthly values for this parameter). The minimum parameter value (**Min: 0.014**) and maximum parameter value (**Max: 0.022**) are shown based on the **Initial Parameter Values**.

The user must designate a **Lower Bound** and an **Upper Bound** for each selected parameter. The lower bound cannot be greater than the minimum parameter value and the upper bound cannot be less than the maximum parameter value. These bounds are used to guide the generation of points in SCE. In SCE, a parameter set is considered a point in N-dimensional space, where N is the number of parameter values in the parameter set. The initial individual values or mean values of the parameter set displayed in Instruction 4-2 are used as one of the points in SCE. The rest of the points are randomly generated by SCE such that each parameter value is within its lower and upper bounds.

After entering the values for lower and upper bounds, the **Actual Range** sampled for the mean in SCE is displayed when **Use the mean value** is chosen (red box in fig. 6). This range is calculated based on the user-defined lower and upper bounds such that no individual values are out of range. **Actual Range** refers to the sampling range used in SCE. The **Actual Range** is calculated as follows:

$$AR_{lb} = \{[(LB + C) * (MEAN_{INIT} + C)] / (min + C)\} - C \qquad (3)$$

and

$$AR_{ub} = \{[(UB + C) * (MEAN_{INIT} + C)] / (max + C)\} - C\ , \qquad (4)$$

where AR_{lb} and AR_{ub} are the lower and upper bounds for the **Actual Range**, LB is the user-defined lower bound, UB is the user-defined upper bound, $MEAN_{INIT}$ is the mean value of initial parameters, min is the minimum initial parameter value, max is the maximum initial parameter value, and C is the absolute value of LB + 10.

When **Use the individual values** is chosen for calibration in Instruction 4-2, each individual value of the highlighted parameter is calibrated. The user is allowed to select which individual values should be calibrated by checking the boxes under **Calibrate?** (blue box in fig. 7). The user is cautioned against using this choice when a parameter is dimensioned by more than 12. After entering the values for lower and upper bounds, the actual range sampled for the individual values in SCE is displayed. The actual range is equal to the user-defined bound when **Use the individual values** is selected as the calibration type.

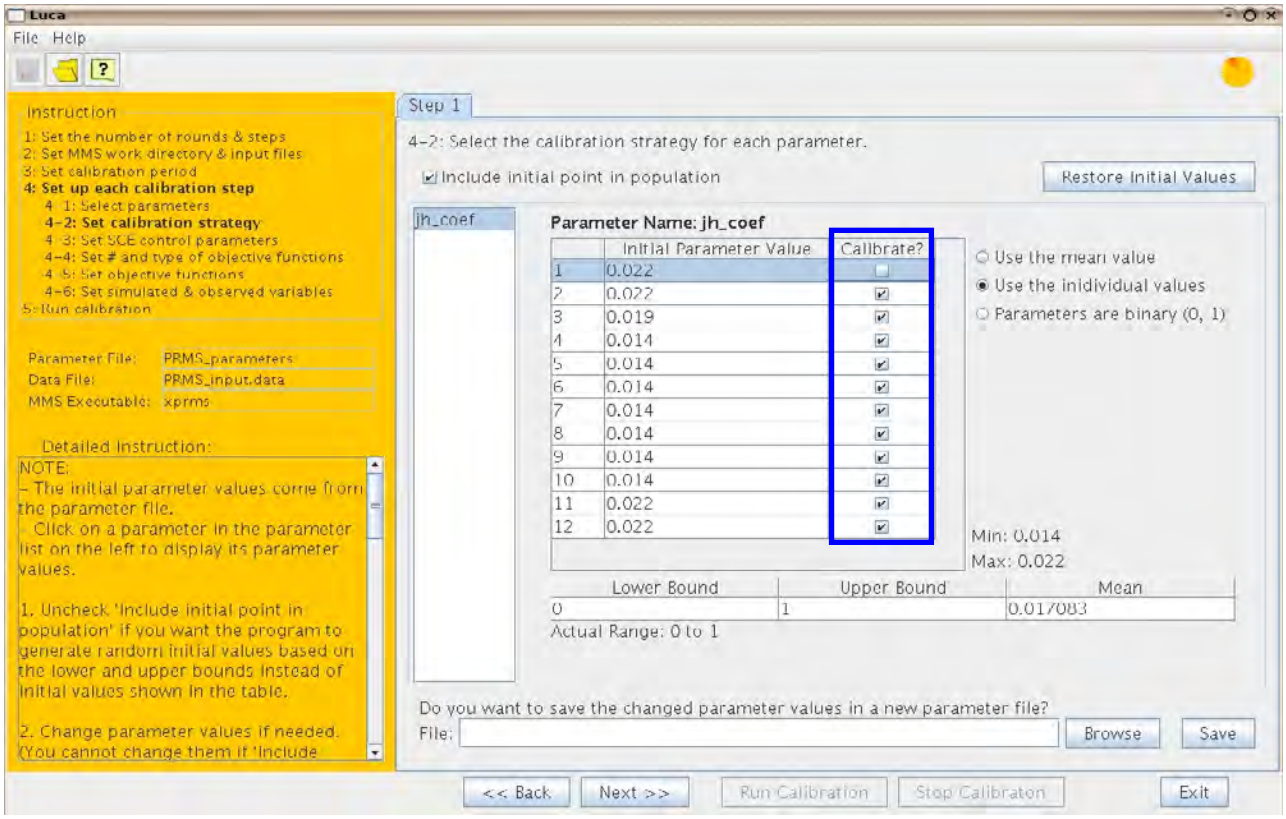

Figure 7. Instruction 4-2 in Luca. "Use the individual values" is chosen.

When **Parameters are binary (0,1)** is chosen for calibration in Instruction 4-2, the parameter values for the highlighted parameter must be either 0 or 1 (see fig. 8). The user is allowed to select which individual values should be calibrated by checking the boxes under **Calibrate?** as shown in figure 8. The **Lower Bound** and **Upper Bound** are (0,1) for this choice. When the SCE-calibrated value is greater than or equal to 0.5, the parameter value is set to 1. Otherwise, it is set to 0. If all parameter values are chosen for calibration, then keep in mind that they may all be set to 1 (or 0) at any time in the SCE process. The user should be familiar with their model. If the model requires one of the binary parameters to be set to 1 (or 0), the user must ensure that the **Calibrate?** box and the **Initial Parameter Value** are set accordingly.

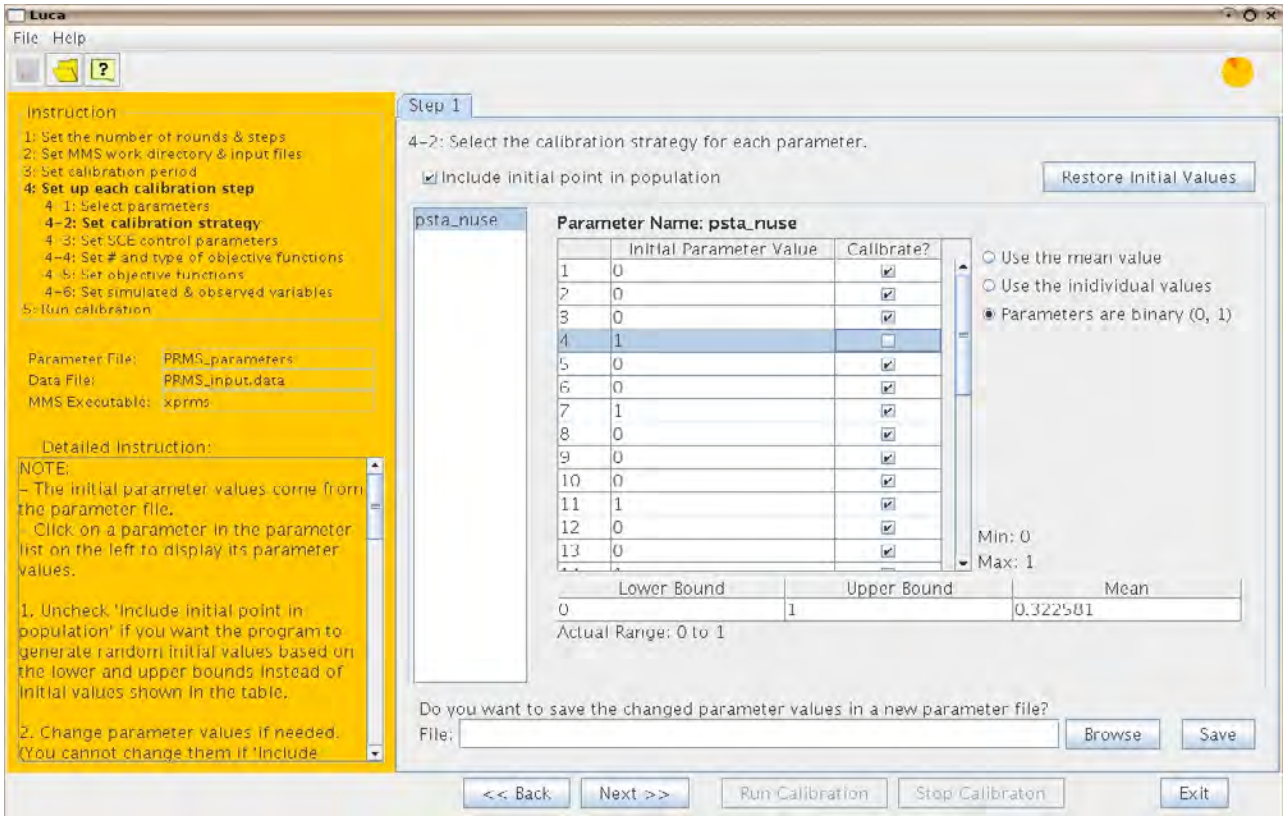

Figure 8. Instruction 4-2 in Luca. "Parameters are binary (0,1)" is chosen.

Instruction 4-3: Set values for Shuffled Complex Evolution control parameters for each step

Instruction 4-3 in Luca, **Set SCE control parameters** for model calibration, is shown in figure 9. The values for the SCE control parameters can be set by either clicking on the step tabs and completing Instruction 4-3 for each step, or by checking the box **Use values in Step 1 for all** (blue box in fig. 9) and filling out instructions for Step 1. The button **Let me modify values!** is made for Linux users (red box in fig. 9). If Luca is running on Linux, the control parameter table may not be editable because of possible inconsistencies between Luca and Java formats. If this happens, click this button to enable editing of the values in the table. The name of the output-parameter file for each step can be set in Instruction 4-3 (see green box on fig. 9).

The SCE control parameters listed in Instruction 4-3 and table 3 are described below in context with the SCE algorithm.

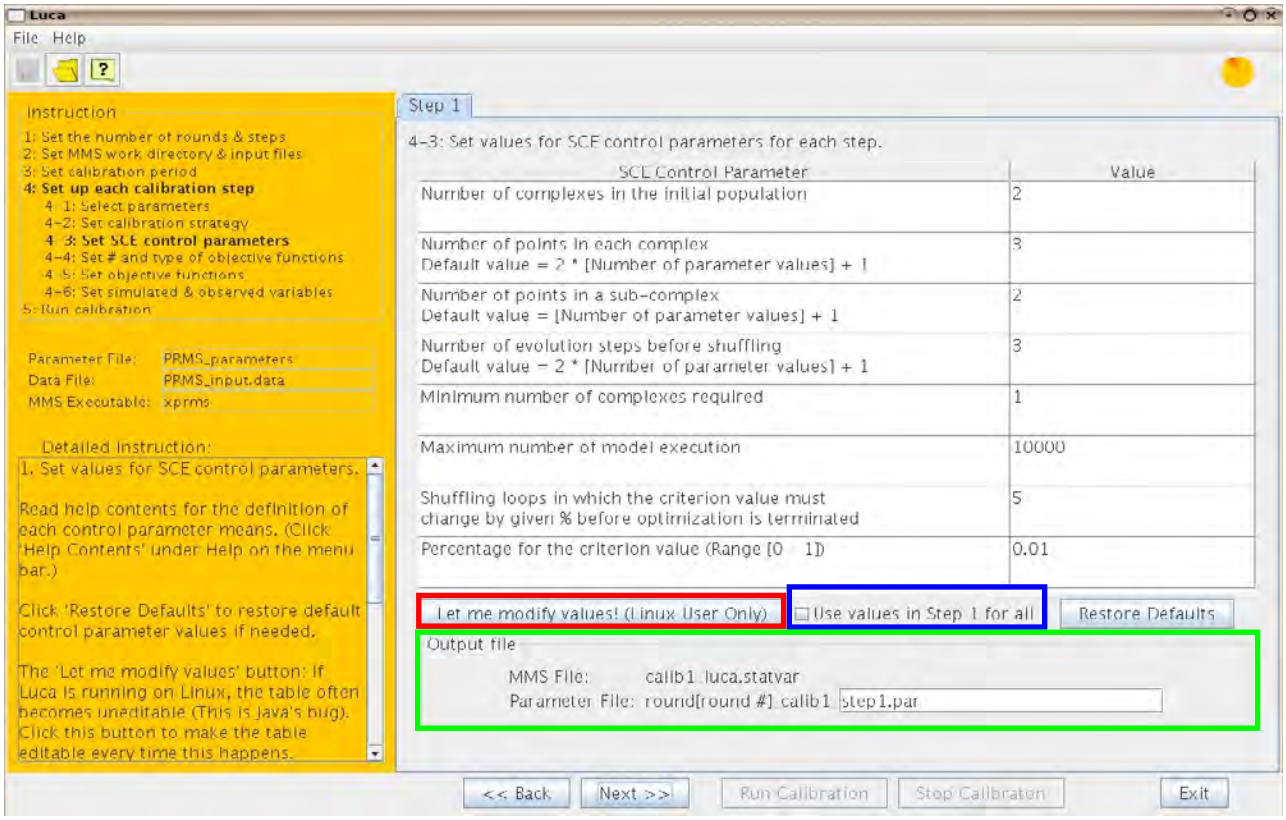

Figure 9. Instruction 4-3 in Luca.

Shuffled Complex Evolution algorithm

Duan and others (1994) provide a detailed description of the SCE algorithm implemented in Luca. The SCE global optimization algorithm, developed by Duan and others (1992), addresses the difficulties in optimization when there are several regions of attraction and multiple local optima in the parameter space. SCE avoids the problem of being trapped in local optima by using a population-evolution-based global search technique, which searches for the optimal solutions from a population of possible solution points, rather than a single point. Figure 10 shows a schematic of the SCE procedure and table 3 describes the SCE control parameters listed in figure 9.

In SCE, the set of parameters to be calibrated is considered a point in N-dimensional space, where N is the number of parameters to be optimized. SCE randomly samples S points in the feasible parameter space. The MMS executable is run with each point (parameter set) and an objective function is calculated. The objective function determines how close the simulation results are to "observed" values. The S objective function values are sorted by increasing order (where lowest is the "best" fit) and then divided into a user-defined number of complexes (P), each containing M points. Each complex is evolved by using the Competitive Complex Evolution algorithm (based on the Nelder and Mead (1965) Simplex Downhill search algorithm). The points in the evolved complexes are combined into a single-sample population. The sample population is sorted by increasing objective function value and shuffled into P complexes.

The shuffling loop is created and repeated until the results of the complex evolution meet one of the following convergence criteria: (1) the number of MMS executions reaches the maximum number of model executions (X in table 3); (2) the percentage change in the best

objective function value of the current shuffling loop and that of several shuffling loops before is less than a specified percentage (OF in table 3); or (3) the points converge into a very small region, which is less than 0.1 percent of the space within the lower and upper bounds of parameters. With each consecutive shuffling loop, the number of complexes created decreases by one (P=P-1). This decrease stops when the number of complexes reaches the minimum number of complexes required (Pmin). The output is the parameter file containing the point (a parameter set) that has the best objective function value.

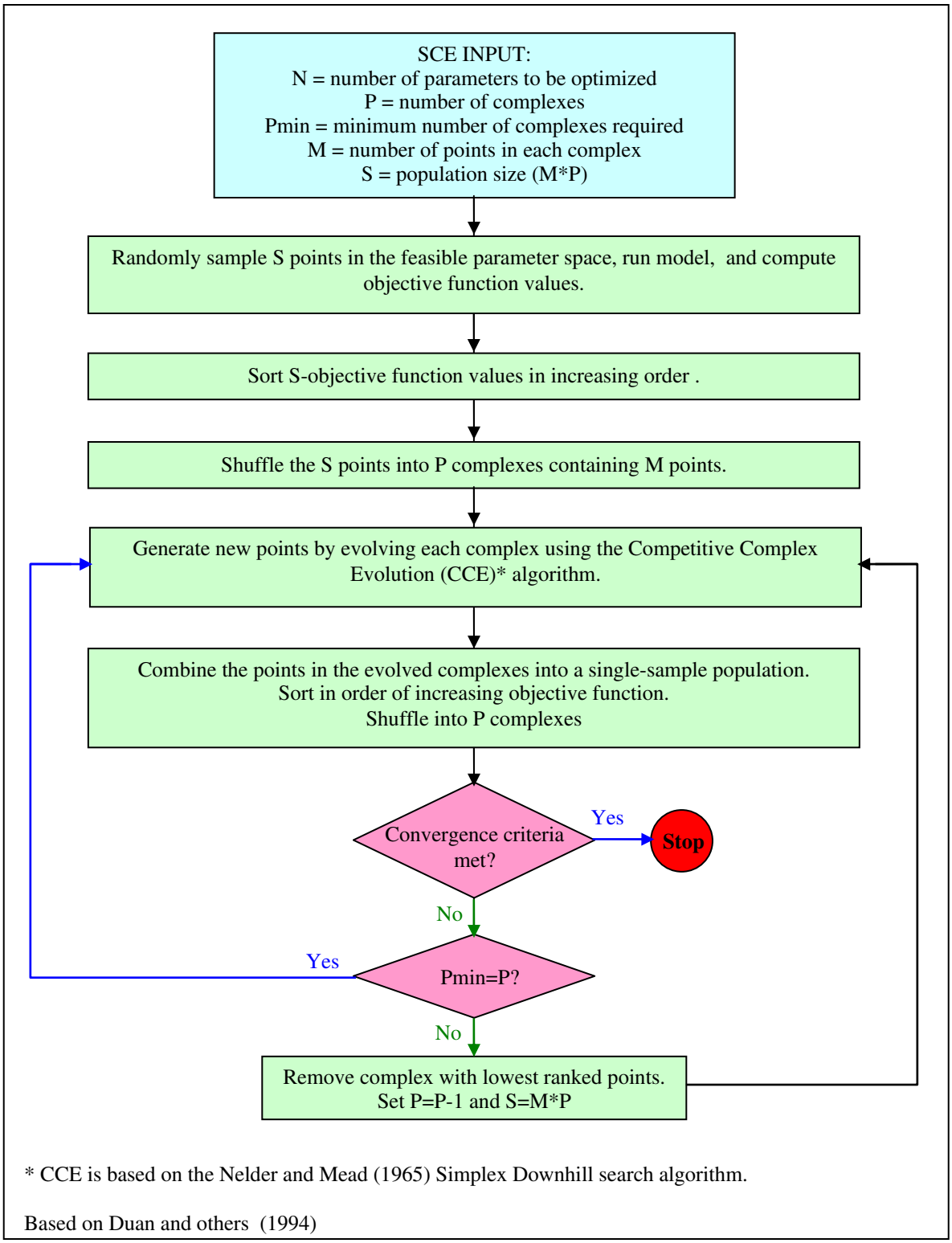

Figure 10. Flowchart of the Shuffled Complex Evolution (SCE) algorithm.

Table 3. Shuffled Complex Evolution control parameters for Luca Instruction 4-3.

[SCE, Shuffled Complex Evolution; Min, minimum; Max, maximum; MMS, Modular Modeling System]

Description	Abbrev	Default	Min	Max
Number of complexes in the initial population. This is constructed from the initial population of points (S). The number decreases by one for every shuffling loop until it reaches the **Minimum number of complexes required**. The program execution time increases as the number of complexes increases.	P	2	1	-
Number of points in each complex. The default value is (2 * N + 1), where N is the total number of values in the parameter set. For example, if the parameter set contains two parameters, both being calibrated with the **Use the mean** option, then N=2. If the parameter set contains 1 parameter, dimensioned by 12 values, and calibrated with the **Use the individual values** option, then N=12.	M	2 * N + 1	Msub	-
Number of points in a sub-complex. This is the number of points randomly selected from a complex to construct a sub-complex.	Msub	N + 1	2	M
Number of evolution steps before shuffling. This is the number of times the evolution steps are performed during each shuffling loop.	ES	2 * N + 1	1	X
Minimum number of complexes required. This is the minimum number of complexes required for the shuffling loops. The number of complexes decreases by one for every shuffling loop until it reaches this minimum number. After that, the number of complexes is kept constant until the end of the program.	Pmin	1	1	P
Maximum number of model executions. A terminating condition for SCE, this is the maximum number of MMS-model executions. If the number of MMS executions reaches this maximum number, SCE terminates.	X	10000	1	-
Shuffling loops in which the objective function value must change by a given percent before optimization is terminated. This is a terminating condition for SCE. If the percent change between the current best objective function value and the best objective function value observed in the specified number of shuffling loops before is less than the **Percentage for the criterion value,** then SCE terminates.	L	5	1	9
Percentage of the objective function. This is a terminating condition for SCE. See the description above.	OF	0.01	0	1

Instruction 4-4: Set the number and type of objective functions for steps

Instruction 4-4, **Set # and type of objective functions**, is shown in figure 11. **The number of objective functions** is set in this instruction and the **Objective Function Type** is chosen. The

user must select a single **Objective Function Type** for each step and the selection should be based on the purpose of the study. Within each step, multiple objective functions (all of the same type) can be defined in **The number of objective functions:** window in figure 11. Note that the weights associated with each objective function are set in Instruction 4-5.

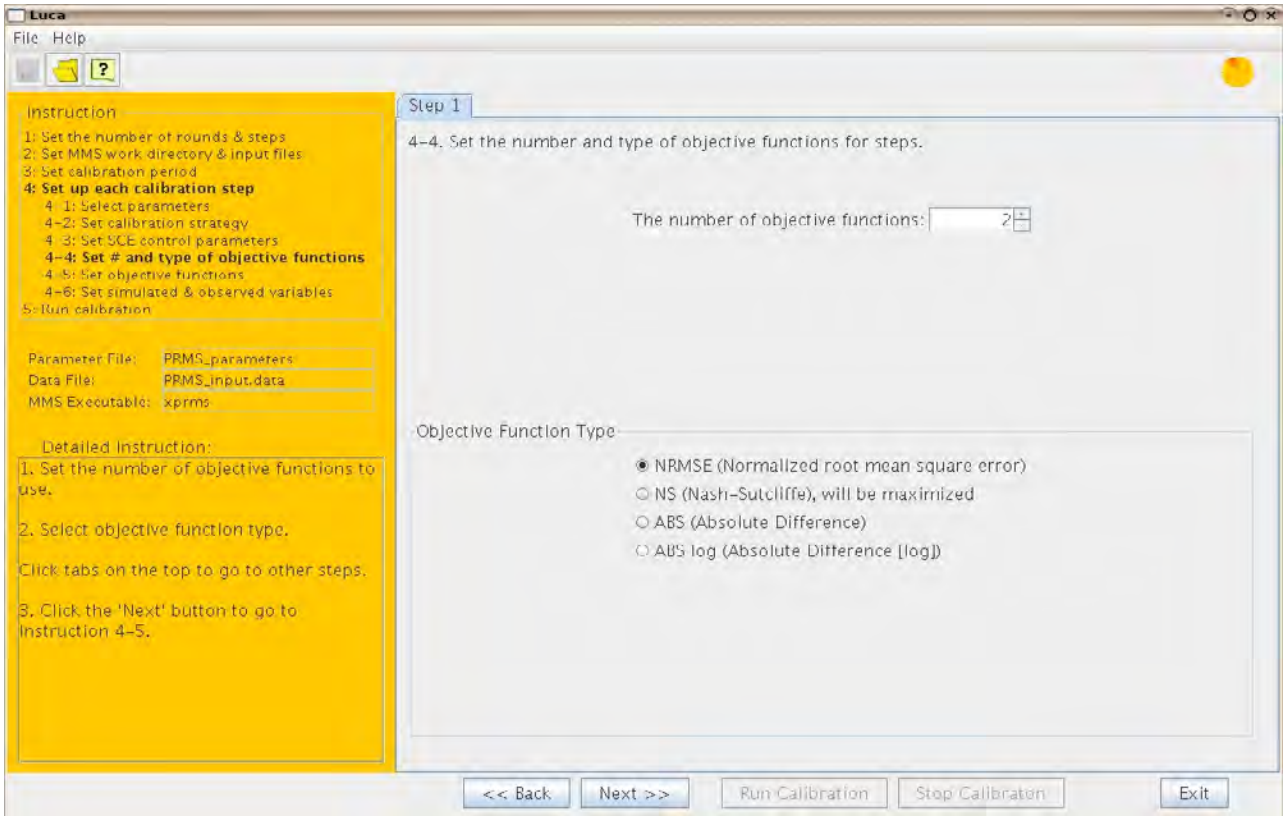

Figure 11. Instruction 4-4 in Luca.

In figure 11, **The number of objective functions** is two and the **Objective Function Type** is NRMSE. The Luca configuration in figure 11 lists four **Objective Function Types** to choose from: **NRMSE (Normalized root mean square error)**; **NS (Nash-Sutcliffe)**; **ABS (Absolute Difference)**; and **ABS log (Absolute Difference [log])**. Additional **Objective Function Types** can be added in future versions of this software. Users may request additional objective functions by contacting the authors. The four **Objective Function Types** are as follows:

NRMSE

The normalized root mean square error is calculated as follows:

$$\mathrm{NRMSE} = \left(\sum_{n=1}^{ntot}(\mathrm{OBS}_n - \mathrm{SIM}_n)^2 \Big/ \sum_{n=1}^{ntot}(\mathrm{OBS}_n - \mathrm{MN})^2\right)^{1/2} \qquad (5)$$

where n is the time step, ntot is the total number of time steps, OBS are the "observed" values, SIM are the simulated values, and MN is the average of the observed. If NRMSE = 0, then the observed values are equal to the simulated values (OBS=SIM). A value of NRMSE > 1 indicates that simulated values are as good as using the average value of all the observed data.

NS

The Nash-Sutcliffe Goodness of Fit statistics (Nash and Sutcliffe, 1970) is computed as follows:

$$NS = 1.0 - \sum_{n=1}^{ntot}(OBS_n - SIM_n)^2 / \sum_{n=1}^{ntot}(OBS_n - MN)^2 \quad . \tag{6}$$

An NS value of one indicates a perfect fit between "observed" and simulated. A value of zero indicates that the fit is as good as using the average value of all the "observed" data. Note this value is maximized in SCE.

ABS

The absolute difference is calculated as follows:

$$\mathrm{ABS} = \sum_{n=1}^{ntot} \mathrm{abs}\left((\mathrm{OBS}_n - \mathrm{SIM}_n)/\mathrm{OBS}_n\right) . \tag{7}$$

ABS log

The absolute difference in the logs is calculated as follows:

$$\mathrm{ABS} = \sum_{n=1}^{ntot} \mathrm{abs}\left((\log \mathrm{OBS}_n - \log \mathrm{SIM}_n)/\log \mathrm{OBS}_n\right) . \tag{8}$$

Instruction 4-5: Set the weight and time step for the objective function

Instruction 4-5, **Set objective functions**, is shown in figure 12. In Luca, the final objective function (OF) value for each step is calculated as follows:

$$\mathrm{OF} = \sum_{i=1}^{nOF}(w_i * \mathrm{OF}_i) \quad , \tag{9}$$

where OF is the final objective function, nOF is the total number of objective functions for a given step, OF_i is the objective function value for the *i*th objective function, and w_i is the weight for the *i*th objective function.

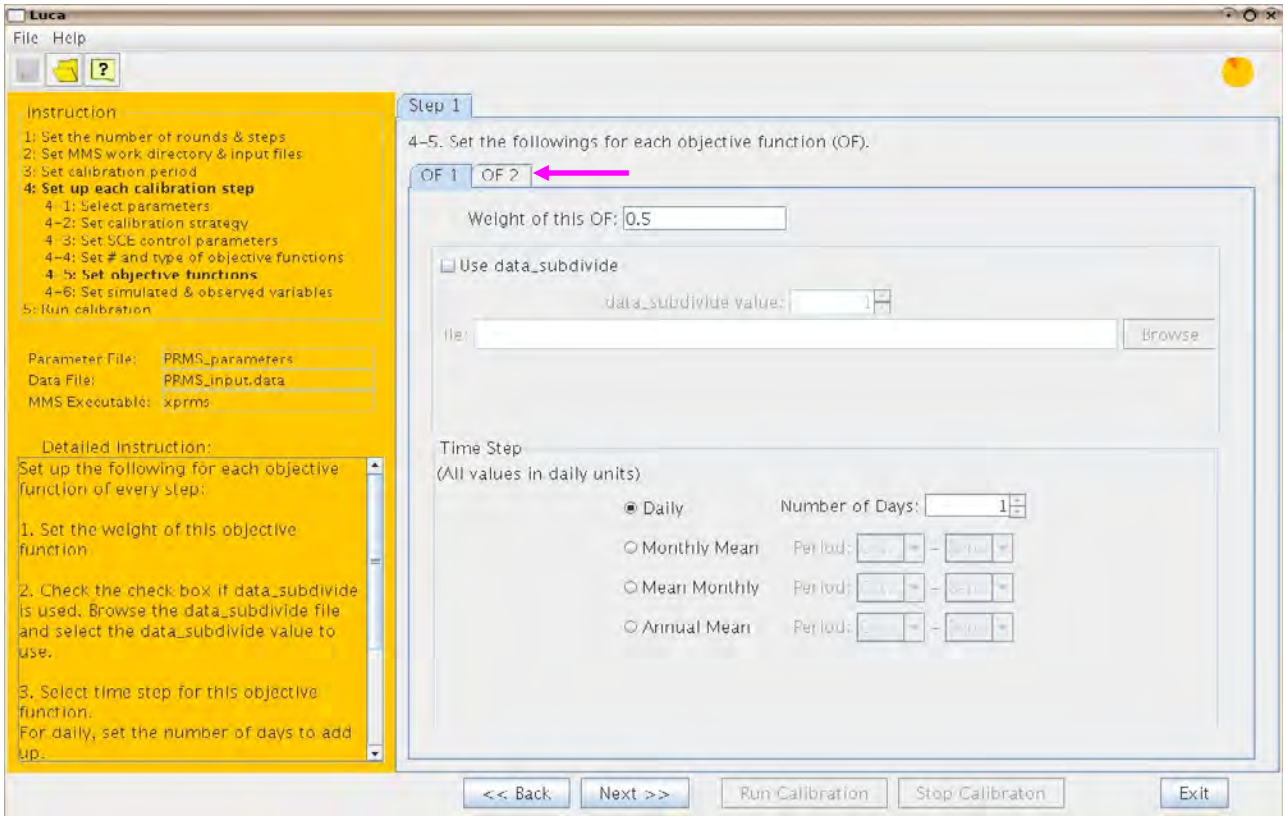

Figure 12. Instruction 4-5 in Luca.

For each step, the **OF** tabs in the panel must be clicked to set the **Weight of this OF**, **Use data_subdivide**, and **Time Step** of each objective function. In figure 12, two tabs are shown (**OF 1** and **OF 2**, indicated by magenta arrow in fig. 12), which means there are two objective functions to set up in Step 1. All instructions for 4-5 must be set for all OF tabs.

The **Use data_subdivide** technique allows the use of a subset of the data for model calibration. If the **Use data_subdivide** box is checked, the user is prompted for a **data_subdivide value** and an input **File**. The format of the data_subdivide **File** is: Year Month Day Value, where Value is used to divide the dates into groups. If Value in the **File** for a given time step is equal to the **data_subdivide value**, then data for that day is used in model calibration. The **data_subdivide value** must be an integer. The data_subdivide **File** must contain data from a time period that is the same or longer than the calibration period, excluding the first year for model initialization. If the "observed" data comes from an external source (see Instruction 4-6), then the external source file must be in the Daily Format (Year Month Day Value).

Use data_subdivide is an effective way to calibrate with user-defined periods within the designated calibration period. For example, the user may only want to calibrate runoff on non-consecutive years or on days with irrigation diversions, peak flows, and/or low flows.

The **Time Step** specified in Instruction 4-5 is used to designate the time step for calculating an objective function value. In the current (2006) Luca configuration, the available time steps are **Daily**, **Monthly Mean**, **Mean Monthly**, and **Annual Mean**.

The **Daily** time-step option has an associated **Number of Days** field. If n days are set for **Number of Days**, then the values, each of which is the sum of n contiguous daily values, are used for objective function calculation. For example, if **Number of Days** is set to three, then a data value

for the date 2001/10/1 is the sum of the data values of 2001/10/1, 2001/10/2, and 2001/10/3. If **Number of Days** is set to one, then objective function calculation is done with individual daily values. If the data_subdivide file is used, then **Number of Days** is automatically set to one.

The **Monthly Mean** (arithmetic mean of individual daily values within a given month), **Mean Monthly** (arithmetic mean of the monthly means of a given month during a specified period of years), and **Annual Mean** have an associated **Period** field. **Period** allows the user to select a period of months for objective function calculation. For example, if the period is set to Oct. – Mar., then the period from October to March of each year are used for objective function calculation, and values from April to September are ignored. If **Period** is set to Oct. – Sept., then all values are used. The calibration period must be at least 2 years (including a 1-year model initialization period) to select **Annual Mean**.

Instruction 4-6: Set simulated and observed variables for objective function for each step

Instruction 4-6, **Set simulated & observed variables**, is shown in figure 13. A **Simulated Variable** is selected from the pull-down **Simulated Variable** list (blue box in fig. 13). The pull-down list contains variables in the MMS Statvar file (see Leavesley and others, 1996 for additional Statvar file information). The corresponding "observed" data must be identified (**Where does the observed calibration data set come from?**), and the source must be indicated as either a **Statvar File** or an **External Source**.

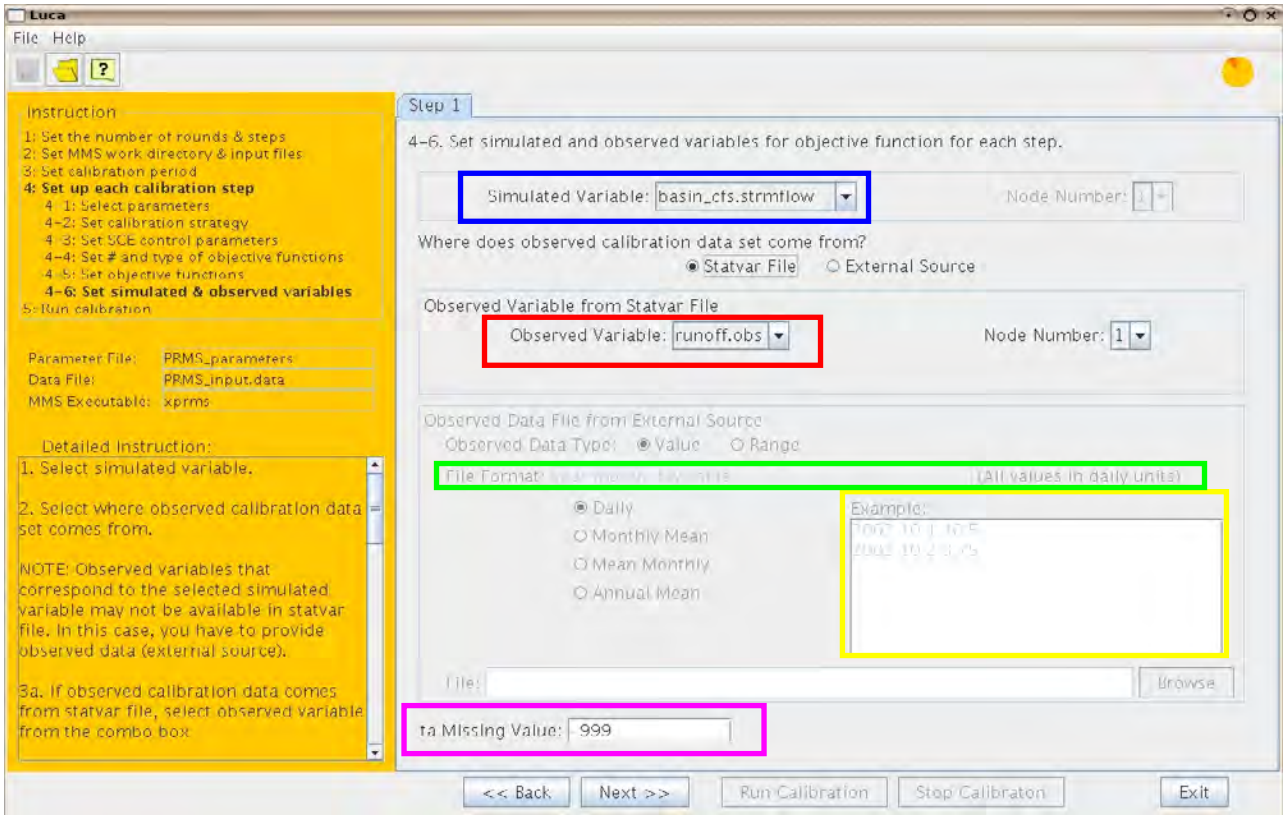

Figure 13. Instruction 4-6 in Luca.

When **Statvar File** is chosen, the "observed" values are contained within the Statvar File and must be chosen from the **Observed Variable from Statvar File** pull-down **Observed Variable** list (red box in fig. 13). This list will contain the "observed" variables present in the Statvar File. In general, when a user sets up an MMS data file, "observed" runoff is included (runoff.obs in fig. 13), which in turn is included in the Statvar file. If the user wants to use "observed" data that is not contained in the Statvar File, then **External Source** must be selected.

When **External Source** is chosen, the user must select the **Observed Data Type** as **Value** or **Range**. The **Browse** button can be used to browse for the "observed" data file that is in the specified format.

If the "observed" data file contains a single value for each date, then select **Value** type. If **Value** type is selected, then the file format of the "observed" data must be specified. This format must be composed of a time step no greater than that chosen as the **Time Step** in Instruction 4-5. For example, if a **Time Step** of **Mean Monthly** was chosen in Instruction 4-5, then the user has the choice of **Daily**, **Monthly Mean**, or **Mean Monthly** for the file format in Instruction 4-6. When the time step is chosen, the **File Format** box will show the appropriate file format and an example of this format will be shown in the Luca **Example** screen (yellow box in fig. 13).

If the "observed" data file contains the lower and upper bound of the "observed" value for each date, then select the **Range** type for **Observed Data Type**. If the **Range** type is selected, then the time step of the data file will automatically default to that chosen in Instruction 4-5. The correct file format will be indicated in the **File Format** box (green box in fig. 13) and an example of this format will be shown in the Luca **Example** screen (yellow box in fig. 13).

When **Range** is chosen as the **Observed Data Type**, error is assessed in the objective function calculations when the simulated value falls outside of this range. The observed value is set to the lower (upper) bound when the simulated value is less (more) than the lower (upper) bound of the range.

The **Data Missing Value** should be filled in if the observed value has a known missing value code (magenta box in fig. 13). If an observed value is equal or less than the **Data Missing Value**, then this observed value and the corresponding simulated value are omitted from objective function calculation. Type "NONE" for the Data Missing Value field to disable this functionality and use all observed values. This technique can be used for any file format (Daily, Monthly Mean, Mean Monthly, and Annual Mean format).

The "observed" data file must contain data covering at least the calibration period excluding a one-year model initialization period. The first year of the calibration period is only for model initialization and not used for objective function calculation. Therefore, the file does not have to contain the data for the first year of the calibration period. The file can contain extra data values of dates other than the calibration period, and the program can still read the data of the specified calibration period correctly. "Observed" values must be in daily units regardless of file formats. For example, an assumption in Luca is that an observed annual mean is calculated by the sum of daily values divided by the total number of days.

Luca Instruction 5: Run the multiple-objective, stepwise calibration

Instruction 5 in Luca, **Run calibration,** is the final instruction. Clicking the **Run Calibration** button at the bottom of the panel in figure 14 (blue box) executes the multiple-objective, stepwise calibration procedure for every step of every round. While Luca is running, the **Program Current Status** panel and the **Best Values and Status of Each Step** panel are updated.

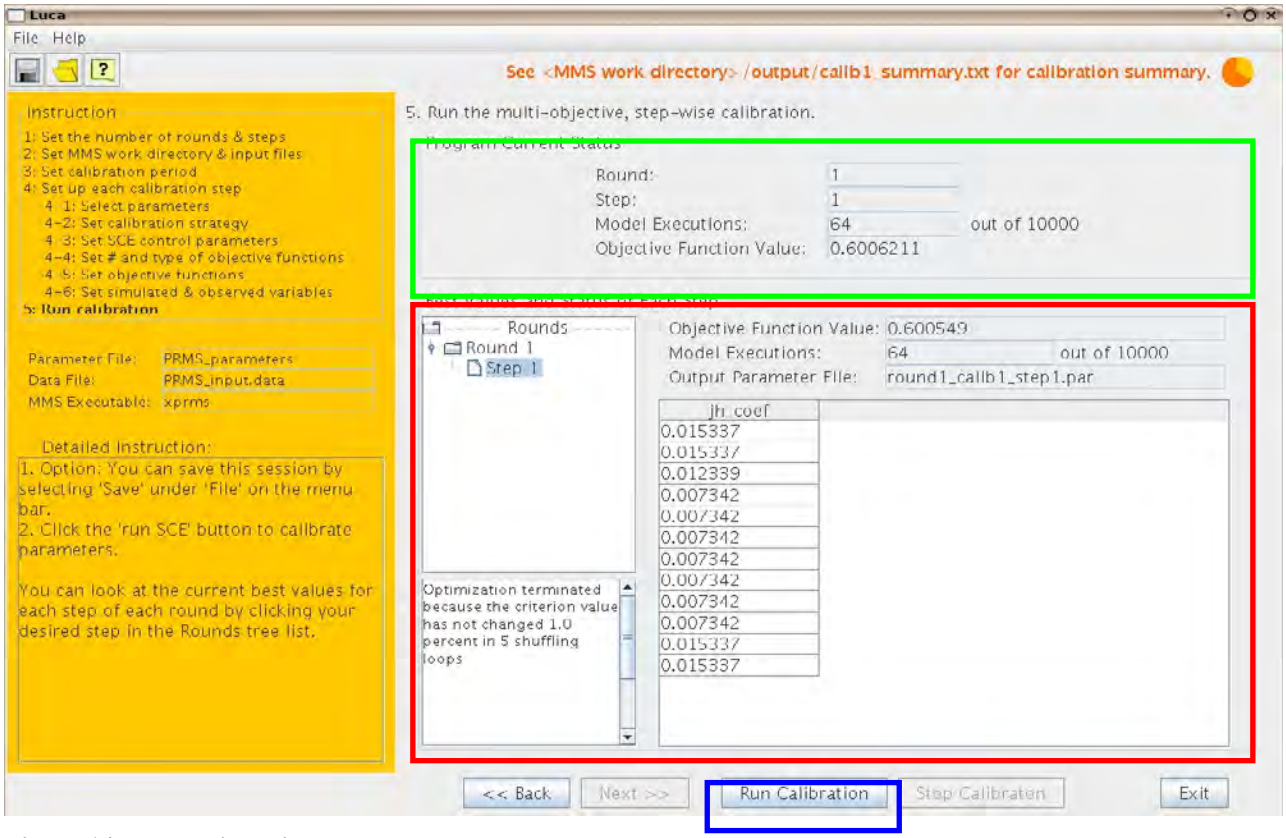

Figure 14. Instruction 5 in Luca.

The fields in the **Program Current Status** panel (green box in fig. 14) display the current **Round**, **Step**, MMS **Model Executions**, and the corresponding **Objective Function Value**.

The **Best Values and Status of Each Step** panel (red box is fig. 14) contains fields associated with each step of each round. The status of each field can be displayed by clicking a node in the tree structure on the left side of the panel. The **Objective Function Value** field shows the current best objective function value of a given step of a round. The best parameter values associated with the current best **Objective Function Value** field are displayed. If SCE has not started a given step of a round, then the initial parameter values are shown. After SCE finishes a step, the final parameter values are shown. The **Model Executions** field displays the current number of MMS model executions for each step of a round. The **Output Parameter File** field displays the name of the output parameter file generated for every step of every round. Luca generates a new output parameter file for each step of each round in the mms_work/input/params directory.

The text box located at the bottom left of Instruction 5 indicates the status of Luca. This field tells the user whether SCE has started, is currently running, or has finished a given step of a round. When Luca is finished, the end statement is displayed in the text box. The end statement explains by one of the following three statements, why the SCE terminated for a given step of a round:

(1) *Optimization search terminated because the limit on the maximum number of model executions, X, was exceeded. Search was stopped at sub-complex M of complex L in shuffling loop K.* X, M, L, and K (in table 3) are substituted with real numerical values. This end statement is

displayed if SCE terminates because the number of model executions in SCE reaches the maximum number of model executions (set in Instruction 4-3).

(2) *Optimization terminated because the criterion value has not changed OF percent in L shuffling loops.* This end statement is displayed if SCE terminates when the percent change in the best objective function value of the current shuffling loop and that of L shuffling loops before is less than OF percent. L and OF are SCE control parameter values set in Instruction 4-3 (see table 3). OF corresponds to the value of **Percentage of the criterion value** (see Instruction 4-3 and table 3), and L corresponds to the value of **Shuffling loops in which the criterion value must change by given percent before optimization is terminated** (see Instruction 4-3 and table 3).

(3) *Optimization terminated because the population has converged into N percent of the feasible space.* N is a normalized geometric mean of parameter ranges. This end statement is displayed if SCE terminates because all points (parameter sets) generated in SCE have converged into in an area that is less than or equal to 0.1 percent of the feasible space (the space within the lower and upper bounds of the selected parameters). The value N is the percentage area into which the points converged.

Summary

This paper introduces the wizard-style graphical user interface, Luca, which is a tool for building a multiple-objective, stepwise, automated calibration procedure to calibrate a hydrological model. The detailed instructions of Luca to build the procedure presented in this paper are designed to help the user get started with Luca

References Cited

Boyle, D.P., H.V. Gupta, and Sorooshian, S., 2000, Toward improved calibration of hydrologic models—Combining the strengths of manual and automatic methods: Water Resources Research, v. 36, no. 12, p. 3663-3674.

Boyle, D.P., H.V. Gupta, and Sorooshian, S., 2003, Multicriteria calibration of hydrologic models—Calibration of watershed models, AGU Water Sciences and Applications, Volume 6: Washington D.C., American Geophysical Union, p. 185-196.

Duan, Q., Sorooshian, S., and Gupta, V., 1992, Effective and efficient global optimization for conceptual rainfall-runoff models: Water Resources Research, v. 28, no. 4, p. 1015-1031.

Duan, Q., Gupta, V.K., and Sorooshian, S., 1993, A Shuffled Complex Evolution approach for effective and efficient global minimization: Journal of Optimization Theory and its Applications, v. 76, no. 3, p. 501-521.

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

Hay, L.E., Leavesley, G.H., Clark, M.P., Markstrom, S.L., Viger, R.J., and Umemoto, M., 2006a, Step-wise, multiple-objective calibration of a hydrologic model for a snowmelt-dominated basin: Journal of the American Water Resources Association, v. 42, no. 4, p. 877-890.

Hay, L.E., Leavesley, G.H., and Clark, M.P., 2006b, Use of remotely sensed snow-covered area in watershed model calibration for the Sprague River, Oregon *in* Joint 8th Federal Interagency Sedimentation Conference and 3rd Federal Interagency Hydrologic Modeling Conference, April 2-6, 2006, Reno, Nevada.

Hay, L.E., Clark, M.P., Pagowski, M., Leavesley, G.H., and Gutowski, W.J., 2006c, One-way coupling of an atmospheric and a hydrologic model in Colorado: Journal of Hydrometeorology, v. 7, no. 4, p. 569-589.

Hogue, T.S., Sorooshian, S., Gupta, H.V., Holz, A., and Braatz, D., 2000, A multi-step automatic calibration scheme for river forecasting models: Journal of Hydrometeorology, v. 1, p. 524-542.

Leavesley, G.H., Lichty, R.W., Troutman, B.M., and Saindon, L.G., 1983, Precipitation- runoff modeling system—User's manual*:* U.S. Geological Survey Water-Resources Investigations Report 83-4238, 207 p.

Leavesley, G.H., Restrepo, P.J., Markstrom, S.L., Dixon, M., and Stannard, L.G., 1996, The modular modeling system, MMS—User's manual: U.S. Geological Survey Open File Report 96-151, 200 p.

Madsen, H., 2000, Automatic calibration and uncertainty assessment in rainfall-runoff modeling *in* ASCE 2000 Joint Conference on Water Resources Engineering and Water Resources planning and management, Minneapolis, July 30-Aug. 2, section 38, chapter 2.

Nash, J.E., and Sutcliffe, J.V., 1970, River flow forecasting through conceptual models part I—A discussion of principles: Journal of Hydrology, v. 10, p. 282-290.

Nelder, J.A., and Mead, R., 1965, A simplex method for function minimization: Computer Journal, v. 7, p. 308-313.

Refsgaard, J.C., 1997, Parameterisation, calibration and validation of distributed hydrological models: Journal of Hydrology, v. 198, p. 69-97.

Turcotte, R., Rousseau, A.N., Fortin, J.P., and Villeneuve, J.P., 2000, A process-oriented, multiple-objective calibration strategy accounting for model structure—Calibration of watershed models, AGU Water Sciences and Applications, Volume 6: Washington D.C., American Geophysical Union, p. 153-164.