EVALUATING THE EFFICIENCY OF A MULTI-CORE AWARE MULTI-OBJECTIVE OPTIMIZATION TOOL FOR CALIBRATING THE SWAT MODEL

X. Zhang, R. C. Izaurralde, Z. Zong, K. Zhao, A. M. Thomson

ABSTRACT. *The efficiency of calibrating spatially distributed hydrologic models is a major concern in the application of these models to understand and manage natural and human activities that affect watershed systems. In this study, we developed a multi-core aware multi-objective evolutionary optimization tool, MAMEO, to calibrate the Soil and Water Assessment Tool (SWAT) model. The efficiency of MAMEO and that obtained with the sequential method were evaluated with data from the Little River Experimental Watershed. By using a 16-core machine, test results showed that calibrating SWAT with the MAMEO method required 80% less time than needed by the sequential method. MAMEO can provide multiple non-dominated parameter solutions in an efficient manner and enable modelers to simultaneously address multiple optimization objectives.*

Keywords. *Calibration, Evolutionary multi-objective optimization, Multi-core aware, Soil and Water Assessment Tool.*

With the availability of spatially distributed data, hydrologists and water resources managers are increasingly using distributed hydrologic models to understand and manage the effects of natural and human activities on watershed systems. The Soil and Water Assessment Tool (SWAT) (Arnold et al., 1998) is a continuous, long-term, distributed-parameter hydrologic model that has been applied worldwide for hydrologic modeling and water resources management. For example, the SWAT model has been incorporated into the U.S. Environmental Protection Agency (USEPA) Better Assessment Science Integrating Point and Nonpoint Sources (BASINS) software package, and SWAT is being applied by the USDA for the Conservation Effects Assessment Project (CEAP) (Gassman et al., 2007). In practical applications for solving water resources problems, the SWAT model's parameters are calibrated to produce model predictions that are as close as possible to observed data. Given the complex structure and large number of parameters of SWAT, a single evolutionary multi-objective optimization (EMO) trial may take several days, weeks, or even longer to reach completion (Zhang et al., 2009b, 2010). Thus, the computational burden is a major concern in the application of SWAT.

Different approaches have been used to improve the efficiency of calibrating SWAT. One such approach utilizes a machine learning technique as a surrogate model to approximate the computationally intensive SWAT. For example, Zhang et al. (2009a) trained artificial neural networks (ANNs) and a support vector machine (SVM) as surrogate models to approximate SWAT. Instead of running SWAT, ANNs or SVM were used to map the response of the objective function into the input parameter space, which could lead to a 20% to 50% time saving for parameter calibration or uncertainty analysis. Another type of approach tries to run SWAT in parallel. For example, Whittaker (2004) used a Beowulf cluster consisting of a server (P4 3.2 GHz dual processor) and 12 computation nodes (P4 2.4 GHz) to calibrate SWAT in parallel using a Monte Carlo algorithm. Later, Confesor and Whittaker (2007) developed a parallel genetic algorithm library (PGAPACK) in FORTRAN that combines the cluster-based parallel computing technique with the non-dominated sorting genetic algorithm II (NSGAII) to calibrate SWAT. This method was later successfully applied by Whittaker et al. (2010) to detect overparameterization and overfitting problems in the automatic calibration of SWAT. Nichols et al. (2011) also successfully tested a cluster-based distributed parallel method to improve the efficiency of executing the Environmental Policy Integrated Climate model (EPIC; Williams, 1995).

Basically, there are two different strategies for parallelizing a sequential algorithm (Pacheco, 2011). The first strategy is based on distributed-memory high-performance computing (HPC) systems, in which a large application is decomposed into a number of parallel tasks and these tasks cooperate and communicate with each other through the

Submitted for review in October 2011 as manuscript number SW 9443; approved for publication by the Soil & Water Division of ASABE in August 2012.

The authors are **Xuesong Zhang,** Research Scientist, and **Roberto Cesar Izaurralde,** Laboratory Fellow, Joint Global Change Research Institute, Pacific Northwest National Laboratory, College Park, Maryland; **Ziliang Zong,** Assistant Professor, Department of Computer Science, Texas State University, San Marcos, Texas; **Kaiguang Zhao,** Research Associate, Nicholas School of the Environment, Duke University, Durham, North Carolina; and **Allison M. Thomson,** Staff Scientist, Joint Global Change Research Institute, Pacific Northwest National Laboratory, College Park, Maryland. **Corresponding author:** Xuesong Zhang, Joint Global Change Research Institute, Pacific Northwest National Laboratory, College Park, MD 20740; phone: 301-314-6706; e-mail: Xuesong.zhang@pnnl.gov.

message passing interface (MPI). The second strategy is based on the shared-memory system (such as a multi-core system), in which the generated parallel tasks collaborate through shared variables in the same memory components. Whittaker (2004) and Confesor and Whittaker (2007) have reported the application of the first strategy to calibrate SWAT, but the multi-core aware technique, which has brought HPC systems into a multi-core era, has not been reported to improve the efficiency of SWAT calibration. Therefore, the major objective of this study was to develop and evaluate the efficiency of a multi-core aware multi-objective evolutionary optimization tool, MAMEO, for calibrating SWAT. The rest of this article is organized as follows. The Materials and Methods section describes the structure and functions of MAMEO, the test watershed, and the methods used to evaluate MAMEO. In the Results and Discussion section, the performance of MAMEO is presented and discussed. Finally, the Conclusions section summarizes the findings of this research.

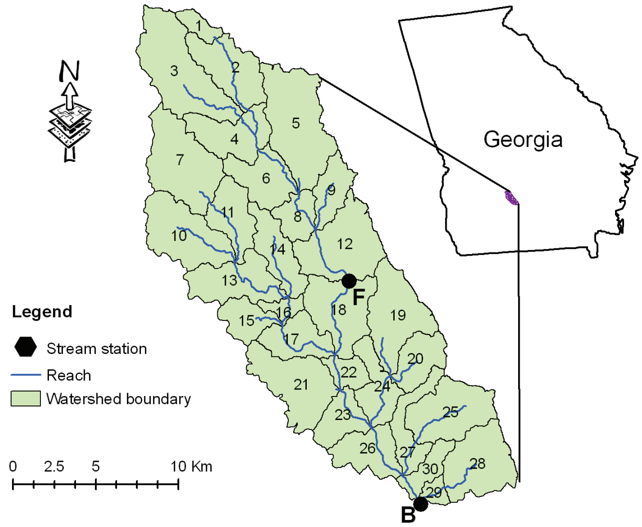


Figure 1. Location of the study area.

Materials and Methods

Study Area Description

The Little River Experimental Watershed (LREW) was selected to test the efficiency of MAMEO for calibrating SWAT. The LREW in southwest Georgia is the upper 334 km^2 of the Little River and has been a subject of long-term hydrologic and water quality research by the USDA-ARS and collaborators (Sheridan, 1997). This region has low topographic relief characterized by broad, flat alluvial floodplains, river terraces, and gently sloping uplands (Sheridan, 1997). Climate in this region is characterized as humid subtropical. Precipitation occurs almost exclusively as rainfall, with an annual mean of 1167 mm. Soils in the watershed are predominantly sandy and sandy loams with high infiltration rates. Since surface soils are underlain by shallow, relatively impermeable subsurface horizons, deep seepage and recharge to regional groundwater systems are impeded (Sheridan, 1997). Land use within the watershed is approximately 50% woodland, 31% row crops (primarily peanut and cotton), 10% pasture, and 2% water. The data used to build the SWAT project were obtained from Bosch et al. (2007). Thirty subwatersheds were delineated within the LREW by using an area threshold of 650 ha. We overlaid the land use map provided by Bosch et al. (2007) and the State Soil Geographic (STATSGO) soil map (http://soils.usda.gov/survey/geography/statsgo/) to delineate HRUs to further characterize the spatial heterogeneity of landscapes within a subwatershed. The establishment of a 5% threshold for both land use and soil areas allowed for the delineation of 129 HRUs. Streamflow data at two observation stations (B and F in fig. 1) were used to test the efficiency of MAMEO.

Multi-Core Aware Multi-Objective Evolutionary Optimization (MAMEO)

The MAMEO tool developed here includes two major components: an EMO and a parallel operator. The Strength Pareto Evolutionary Algorithm 2 (SPEA2), developed by Zitzler and Thiele (1999), was used for multi-objective optimization based on the test and comparison of several EMOs for optimizing SWAT by Zhang et al. (2010). The parallel operator was based on the Python multiprocessing function, which can simultaneously send multiple processes or threads to the cores in a computer (http://docs.python.org/library/multiprocessing.html). A brief description of SPEA2 and its parallel execution in MAMEO is presented below.

The multi-objective optimization problem can be defined as follows (refer to the Nomenclature section for symbol definitions). Find the parameter solution $\boldsymbol{x}^*$ that optimizes the objective function vector $\boldsymbol{F}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_m(\boldsymbol{x})]$. An objective function vector $\boldsymbol{F}(\boldsymbol{x}') = [f_1(\boldsymbol{x}'), f_2(\boldsymbol{x}'), ..., f_m(\boldsymbol{x}')]$ is said to dominate another objective function vector $\boldsymbol{F}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_m(\boldsymbol{x})]$, which is denoted by $\boldsymbol{F}(\boldsymbol{x}') \succ \boldsymbol{F}(\boldsymbol{x})$, if (Zitzler and Thiele, 1999):

$$\begin{aligned} &\forall i \in \{1,\ 2, \ldots,\ m\}, f_i(\boldsymbol{x}') \geq f_i(\boldsymbol{x}) \\ &\text{Ê}\ \exists i \in \{1,\ 2, \ldots,\ m\}, f_i(\boldsymbol{x}') > f_i(\boldsymbol{x}) \end{aligned} \tag{1}$$

If the objective function vector $\boldsymbol{F}(\boldsymbol{x}^*)$ of a parameter solution $\boldsymbol{x}^* \in \Omega$ is not dominated by the other objective function vectors of the parameter solutions in the feasible parameter space, then $\boldsymbol{x}^*$ is taken as a Pareto optimal parameter solution. The Pareto optimal set (P^*) is defined by the set of parameter solutions that are not dominated by other parameter solutions. The objective function vectors corresponding to the Pareto optimal set comprise the Pareto front (PF^*). SPEA2 requires users to initialize a population of parameter solutions (P_t) and an empty external archive (P'_t), which are evolved using fitness assignment, environmental selection, and offspring production operations. In the fitness assignment operator, each individual in P_t and P'_t is assigned to a fitness. The environmental selection operator is used to copy all Pareto optimal chromosomes in P_t and P'_t to P'_{t+1}. If the size of P'_{t+1} exceeds N', then P'_{t+1} is reduced by means of truncating the non-dominated chromosomes with less fitness. Otherwise, if the size of P'_{t+1} is less than N', then P'_{t+1} is filled with the best dominated chromosomes in P_t and P'_t. In the offspring reproduction operator, genetic algorithms (GA) (Goldberg, 1989) are used to reproduce N parameter solutions in P_{t+1} using P'_{t+1}. The objective functions of each member in P_{t+1} are then calculated. The fitness assignment, environmental selection, and offspring production are repeated until a maximum number of iterations (T) is reached. In this study, the binary tournament selection, simulated binary crossover, and polynomial mutation operators were employed within the GA. Detailed description of SPEA2 is provided by Zitzler and Thiele (1999).

Figure 2 shows the schematic framework of MAMEO to execute SPEA2 in parallel. First, we initialize a population of parameter solutions (P_t) and an empty external archive (P'_t), and create N folders. Each folder corresponds to a unique member in P_t and contains the files used for running SWAT and calculating objective functions. Second, for the first n members in P_t, a parallel operator is called to simultaneously modify parameters, run SWAT, and calculate objective functions using the n cores. This procedure is fulfilled using the Python multiprocessing function and is repeated for the rest of the members in P_t until all members are treated. Next, SPEA2 and the parallel operator are iteratively executed to create new P_{t+1} and P'_{t+1} and run SWAT until the maximum number of iterations (T) is reached.

To achieve the aim of calibrating SWAT in parallel, substantial expansion of the original SPEA2 algorithm is required to interface it with the SWAT model and the multiprocessing capability of Python. MAMEO contains several key modules:

- The main program that implements SPEA2 in a parallel means, as explained in figure 2.
- The GetOri module to read in the default parameter values of each HRU (e.g., CN2 and SOL_AWC), which are the baseline for adjusting parameter values during calibration.
- The COF module to read in the basic settings of the objective functions to be optimized, including objective function type (e.g., Nash-Sutcliffe efficiency (NSE), percent of bias (PBIAS), and ratio of the root mean square error to the standard deviation of measured data (RSR), Moarisi et al., 2007), locations of streamflow stations to be calibrated (e.g., subbasin or reach number), and the files that contain the observed data.
- The RS module to execute the SWAT model for the specified simulation period and print out the simula-

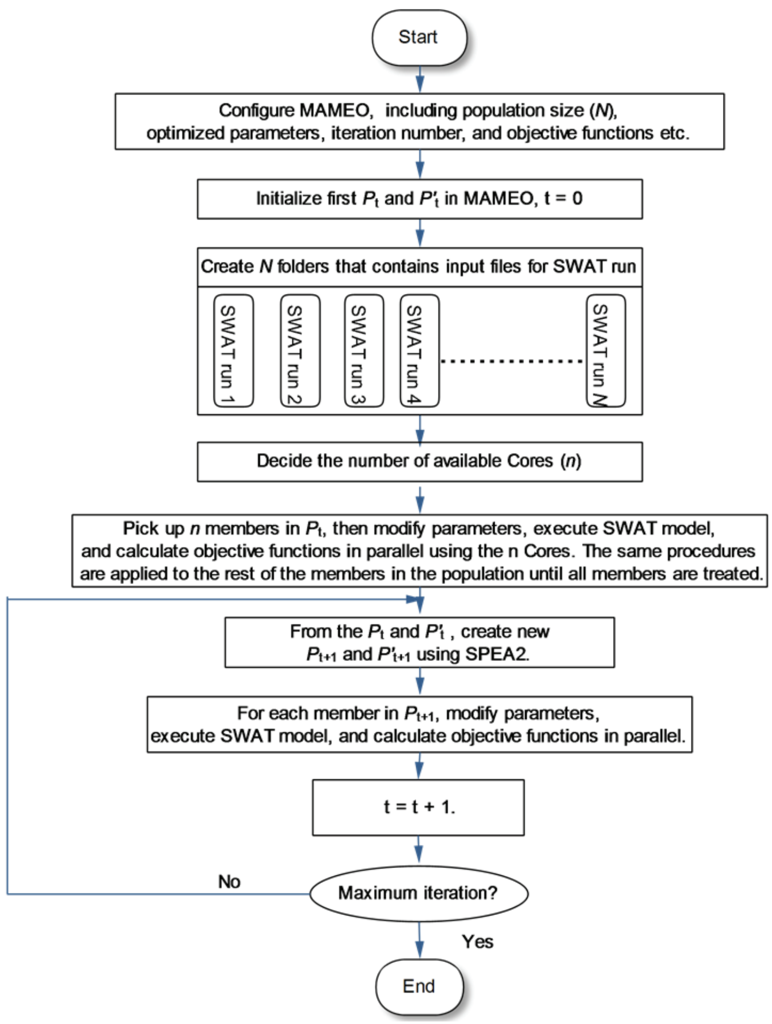


Figure 2. Parallel execution of evolutionary multi-objective optimization of SWAT in a multi-core aware environment.

tion results at a specific subwatershed or reach that will be compared with observations.

- The MP module to adjust parameters of each HRU, subwatershed, and reach during each iteration of SPEA2.
- The OutAnalysis module to generate simulation results by using the non-dominated solutions produced by SPEA2. For example, this module will create one folder for each non-dominated parameter set and run SWAT for both calibration and validation periods to derive results for further examination.

MAMEO contains about 3,000 lines of code, of which only 300 lines are related to the SPEA2 algorithm. The Python codes of SPEA2 in MAMEO were translated from the Visual Basic codes that have been tested by Zhang et al (2010) for multi-objective optimization of SWAT. MAMEO uses binary tournament selection (Goldberg, 1989), simulated binary crossover, and polynomial mutation (Deb et al., 2002) in SPEA2 to evolve and optimize parameter solutions. Following Zhang et al. (2010), key settings of SPEA2 were as follows: probability of crossover of 1.0, crossover distribution index of 15, mutation distribution index of 20, probability mutation of 1/*D*, and real variable representation.

Performance Evaluation of MAMEO

The efficiency of MAMEO was evaluated on a 16-core machine and compared with sequential execution of SPEA2 using only one core on the same machine. Figure 3 shows the architecture of the 16-core machine with four quad-core Intel Xeon processors. The clock speed of the quad-core processors is 3 GHz. Each processor has an L2 cache of 8 MB. The total size of RMA is 24 GB. The operating system is a 64-bit Linux.

The speedup coefficient (Scott et al., 2005) was used to evaluate the efficiency of MAMEO on this 16-core machine. The objective functions to be optimized are Nash-Sutcliffe efficiency (NSE) (Nash and Sutcliffe, 1970) at the two monitoring stations (B and F in fig. 1). Optimal parameter solutions obtained by SPEA2 can vary substantially from one trial to another because the optimization process relies on intensive random number samplings. Usually, multiple trials are preferred to obtain the final optimization results for analysis (Ali et al., 2005). Therefore, we used ten trials of SPEA2 to derive the optimization results for each test case. The baseline process time was estimated by averaging the results from ten sequential SPEA2 executions using only one core. Then the speedup coefficients of using 2, 4, 8, and 16 cores on this 16-core machine were calculated by dividing the baseline process time by the process time consumed by MAMEO with various numbers of cores. Larger speedup values indicate more time saving or higher efficiency. In both sequential and parallel SPEA2 executions, the population size was set to 80 and the maximum iteration was 100. The selection of the population size and maximum iteration was based on the finding by Zhang et al. (2010) that an execution of SPEA2 in multiple trials with a relatively small number of model runs slightly outperforms running SPEA2 once with long iterations. We also considered the even distribution of computational loads across the 16 cores as a factor in determining the population size.

The sensitivity of speedup to the number of simulation years was assessed. As most SWAT studies use two to eight years for model calibration (Gassman et al., 2007), we employed three simulation periods in our sensitivity test: two years (1998-1999), four years (1998-2001), and eight years (1997-2004). Along with previous examinations of SWAT in the LREW (Van Liew et al., 2007; Zhang et al., 2009c), we included the eleven parameters listed in table 1 in our calibration of streamflows at stations F and B. The ranges of these parameters were determined based on the values reported by Neitsch et al. (2005), Van Griensven et al. (2006), and Van Liew et al. (2007).

Results and Discussion

Efficiency Achieved by MAMEO

With the multi-core aware parallel computing capability, the time consumed by executing the SPEA2 algorithm was significantly reduced (fig. 4). For example, for the 8-year simulation scenario, MAMEO consumed about 153 s for an iteration of SPEA2 with 16 cores, which represents about

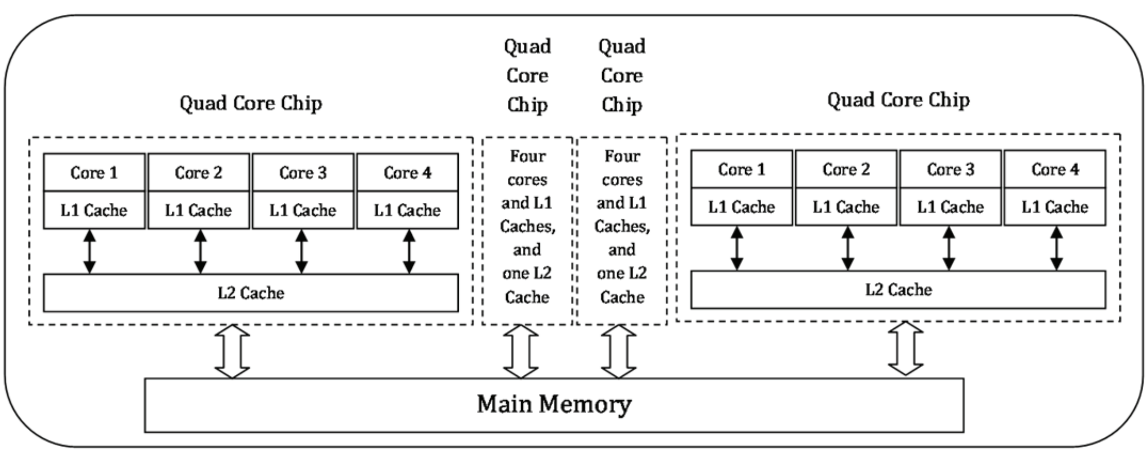
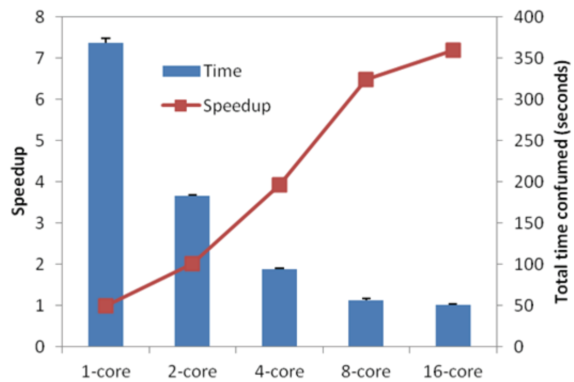


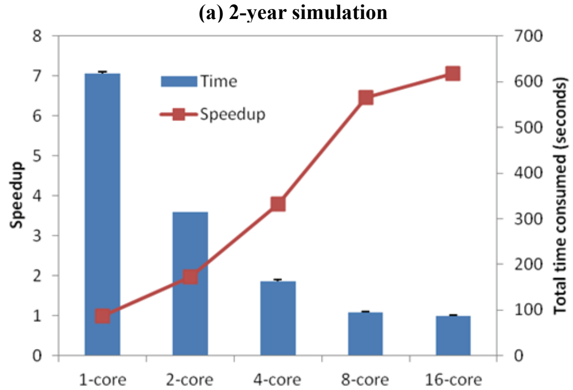
Figure 3. Schematic diagram of the architecture of the 16-core machine with four quad-core Intel Xeon processors.

Table 1. Parameters for calibration in the SWAT model.

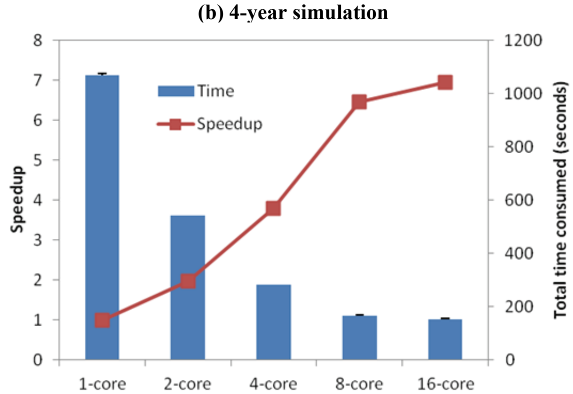
Code	Parameter	Description	Range
1	CN2	Curve number	±10%
2	ESCO	Soil evaporation compensation factor	0 to 1
3	SOL_AWC	Available soil water capacity	±20%
4	GW_REVAP	Groundwater re-evaporation coefficient	0.02 to 0.2
5	REVAPMN	Threshold depth of water in the shallow aquifer for re-evaporation to occur (mm)	0 to 500
6	GWQMN	Threshold depth of water in the shallow aquifer required for return flow to occur (mm)	0 to 5000
7	GW_DELAY	Groundwater delay (days)	0 to 50
8	ALPHA_BF	Base flow recession constant	0 to 1
9	RCHRG_DP	Deep aquifer percolation fraction	0 to 1
10	CH_K2	Effective hydraulic conductivity in main channel alluvium (mm h^{-1})	0.01 to 150
11	SURLAG	Surface runoff lag coefficient (day)	0 to 10



(a) 2-year simulation



(b) 4-year simulation



(c) 8-year simulation

Figure 4. Total time consumed by the execution of SWAT in one iteration (80 SWAT runs) of SPEA2 with different numbers of cores and the corresponding speedup achieved.

15% of the time consumed by running SWAT sequentially with only one core. Similar results were obtained for the 2-year and 4-year simulation scenarios. With a total of 100 iterations, one trial of the SPEA2 algorithm took about 29.7 h for the 8-year simulation scenario. By using 16 cores, only 4.3 h were needed. These results show the promise of using multi-core aware techniques to improve the efficiency of calibrating the SWAT model.

Table 2. Speedup achieved by three parallelized components of MAMEO.

Number of Simulation Years	Number of Cores	Speedup of MP	Speedup of RS	Speedup of COF	Total Speedup
2	2	2.11	1.98	2.041	2.02
	4	4.05	3.85	4.02	3.93
	8	6.74	6.14	7.35	6.48
	16	7.22	6.70	9.02	7.18
4	2	1.96	1.96	1.99	1.97
	4	3.83	3.76	3.91	3.80
	8	6.04	5.77	6.55	6.46
	16	7.02	6.64	8.94	7.07
8	2	1.94	1.97	1.99	1.97
	4	3.76	3.83	3.70	3.80
	8	6.15	6.34	7.13	6.45
	16	6.77	6.64	8.87	6.95

Notably, the speedup achieved by increasing the number of cores increased linearly. For the 2-core, 4-core, and 8-core scenarios, the speedup was approximately the same as or slightly less than the number of cores used (fig. 4 and table 2). However, when we used 16 cores, the speedup did not increase by a factor of 2 as compared with that of using 8 cores. Instead, a speedup of around 7 was achieved (table 2). This, to some extent, is because the shared caches may become the performance bottleneck. In the multi-core architecture shown in figure 3, each core manages its own level 1 cache and shares an L2 cache. This sharing of an L2 cache could lead to performance degradation when two or more threads are co-running with cache-intensive operations.

The number of simulation years can slightly influence the efficiency of the multi-core aware parallel execution of SWAT. Overall, with the increase in the number of simulation years, the speedup achieved decreased. For example, for the 16-core scenario, speedup decreased from 7.18 for the 2-year simulation scenario to 7.05 and 6.95, respectively, for the 4-year and 8-year simulation scenarios. Similar results were observed for the 2-core, 4-core, and 8-core scenarios. Input/output (I/O) delay may be the reason for the degradation of MAMEO's performance. No matter how many cores are available, a task cannot be executed until all data needed have been fetched to the memory. Each SWAT model run requires intensive reading and writing of large amounts of data from and into text files. A longer simulation period means a more intensive I/O operation.

As MAMEO contains three major parallelized components, i.e., Modify Parameters (MP), Run SWAT (RS), and Calculate Objective Functions (COF), we further analyzed the efficiency achieved by each single component (table 2). COF attained the highest speedup, while RS achieved the least. The MP component achieved slightly higher efficiency than RS. The least speedup achievement of the RS component, to some extent, can be explained by the intensive operations of reading and writing files during the execution of SWAT, causing I/O delay. The MP component only needed to process a proportion of the files that were read or written by SWAT; therefore, it achieved higher efficiency than RS. The COF component only involved processing 2× the number of objective functions (calculating an objective function required reading one observed and one simulated time series of streamflow data from two separate files), which may be the reason that it achieved the highest efficiency among the three components.

To further test the efficiency of MAMEO for a more complex and time-consuming SWAT execution case, we created one more scenario in the LREW by increasing the number of simulation years to 20 and HRUs to 2,019. This scenario was created through re-delineating HRUs. The land use and soil data were overlaid with a slope map with four slope classes (0-3, 4-6, 6-10, and >10), and an area threshold of 0 was adopted for all layers when delineating HRUs.

For this complex test case, MAMEO achieved speedup of 1.9, 3.66, 5.91, and 6.32, respectively, for the 2-core, 4-core, 8-core, and 16-core executions (fig. 5). The efficiency achieved was less than that obtained for the test cases shown in table 2 and figure 4. The reason may be that this test case involves more I/O and cache-intensive operations. Notably, the 100 iterations (80 SWAT runs for each iteration) of SPEA2 for this complex test case took about 218 h. By using 2, 4, 6, 8, and 16 cores, the time saved for one trial of SPEA2 was 103, 158, 181, and 184 h, respectively. The computational efficiency of MAMEO allows SWAT modelers to save several days for one trial of a SPEA2 execution to optimize the SWAT model. SPEA2 and many other EMOs are based on random sampling; it is preferable to execute multiple trials of SPEA2 and combine the optimization results to provide a wide range of candidate parameter solutions that are simultaneously addressing multiple objectives. As such, the parallel computing capability of MAMEO is expected to help fully utilize available computational resources and significantly save computational time during the calibration of SWAT.

Overall, the aforementioned observations demonstrate the potential of a multi-core platform for efficient parallel calibration of the SWAT model. The capacity of shared caches and I/O delay may affect the efficiency of MAMEO. In MAMEO, the SWAT executable program is called externally, which limits the flexibility of improving the I/O operations in the SWAT model. In order to further enhance the efficiency of parallel calibration of SWAT, its structure may need to be re-engineered to optimize the I/O operations.

Sensitivity of MAMEO to the Number of Cores

While the efficiency of optimizing SWAT is the major objective for developing MAMEO, it is also important to ensure that the performance of SPEA2 is not deteriorated. SPEA2 involves random sampling of parameter values, so the results obtained by one trial are stochastic and cannot be used to accurately evaluate the algorithm's performance (Ali et al., 2005). The simulations from ten trials were combined to derive optimization results to compare the performance of different methods. The Pareto fronts attained by the sequential, 8-core, and 16-core SPEA2 executions are shown in figure 6 for the 4-year simulation scenario. These three SPEA2 executions produced very similar Pareto fronts in terms of both extent and optimal solutions. Further analysis showed that the ranges of the non-dominated parameter solutions respectively obtained by the sequential, 8-core, and 16-core SPEA2 executions were very close to each other (table 3). Similar results were also observed for the 2-year and 8-year simulation scenarios and the 2-core and 4-core execution schemes. This indicates that the integration of SPEA2 with the parallel multi-core aware technique did not impair the performance of SPEA2 and allowed for consistent performance in terms of finding Pareto optimal parameter solutions for SWAT.

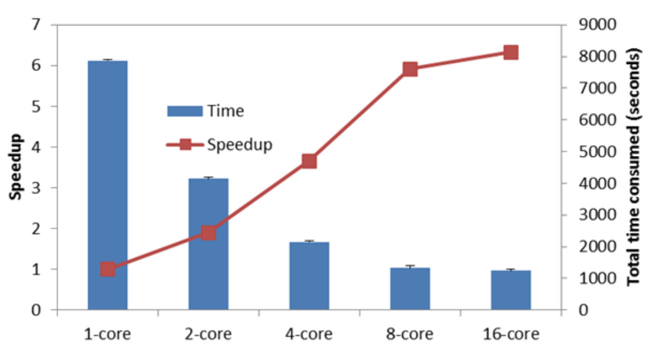


Figure 5. Total time consumed by the execution of SWAT in one iteration (80 SWAT runs) of SPEA2 with different numbers of cores and the corresponding speedup achieved for the 20-year simulation scenario with 2019 HRUs in the LREW.

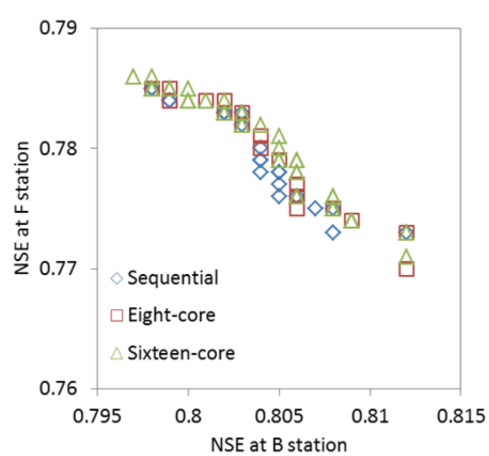


Figure 6. Pareto fronts attained by different SPEA2 executions for the test case with the 4-year simulation. (Results from ten trials of each SPEA2 execution were combined to derive the corresponding Pareto front.)

Table 3. Ranges of non-dominated parameter solutions obtained by different SPEA2 executions.

Parameter	Sequential Execution		8-Core Execution		16-Core Execution	
	Max.	Min.	Max.	Min.	Max.	Min.
Surlag	0.502	0.5	0.502	0.5	0.502	0.5
ESCO	0.932	0.889	0.932	0.903	0.919	0.902
CN	-0.097	-0.1	-0.09	-0.1	-0.09	-0.1
GW_DELAY	2.552	1.818	2.552	1.818	2.559	1.796
ALPHA_BF	0.999	0.943	0.999	0.953	0.999	0.939
GW_REVAP	0.102	0.077	0.102	0.077	0.102	0.084
GWQMN	268.01	265.464	267.832	266.977	267.832	266.978
RCHRG_DP	0.003	0	0.007	0	0.003	0
REVAPMN	196.163	191.272	194.237	142.805	196.07	191.078
SOL_AWC	0.2	0.149	0.2	0.167	0.2	0.154
CH_K2	149.964	149.543	149.937	149.543	149.946	149.556

Potential Use of MAMEO in SWAT Calibration

The major strength of MAMEO is to provide multiple non-dominated parameter solutions in an efficient manner. This would allow modelers to explore the tradeoffs of multiple parameter sets with respect to multiple performance metrics and select one parameter solution or multiple parameter solutions to conduct watershed simulations (Zhang et al., 2011). As it is beyond the scope of this study to extensively explore the use MAMEO in SWAT modeling, here we briefly discuss its potential use in supporting model selection.

Moriasi et al. (2007) recommended the use of multiple performance measures, including NSE, PBIAS, and RSR, to evaluate whether SWAT is satisfactorily calibrated. The non-dominated parameter solutions obtained by MAMEO during a calibration stage provide multiple candidates that can be selected by using those performance metrics. Here we used the 51 non-dominated parameter solutions obtained using the 16-core SPEA2 execution and the 4-year calibration period (1998-2001) to illustrate how MAMEO could be used to assist in model selection. For the calibration period, all 51 parameter solutions have NSE >= 0.75, PBIAS =< 10%, and RSR =< 0.5 at both stream stations and can be taken as providing "very good" performance for streamflow simulation (Moriasi et al., 2007; here, we directly inherited the performance rating criteria provided for a monthly time step to evaluate our simulations at a daily time step). However, when validated for a 3-year validation period (2002-2004), only one parameter solution can simultaneously provide "good" performance (NSE >= 0.65, PBIAS =< 15%, and RSR =< 0.6) at both stream gauges (fig. 7). This parameter set had an NSE of 0.68, RSR of 0.57, and PBIAS of 12.1% at station B and an NSE of 0.64, RSR of 0.6, and PBIAS of -3.7% at station F during the validation period. This parameter set with "good" performance is preferred to the others in terms of assessing watershed management activities and/or climate change impacts. When auxiliary information is available (e.g., soil moisture and evapotranspiration), modelers could include extra dimensions in MAMEO during calibration to simultaneously optimize multiple hydrological variables to achieve a better characterization of watershed processes. In addition, the simulations from the multiple non-dominated parameter solutions can serve as candidates for the Bayesian model averaging (BMA) method, which has been shown as an effective method for estimating uncertainty of SWAT modeling, as exemplified by Zhang et al. (2009c), Sexton et al. (2010), and Strauch et al. (2012).

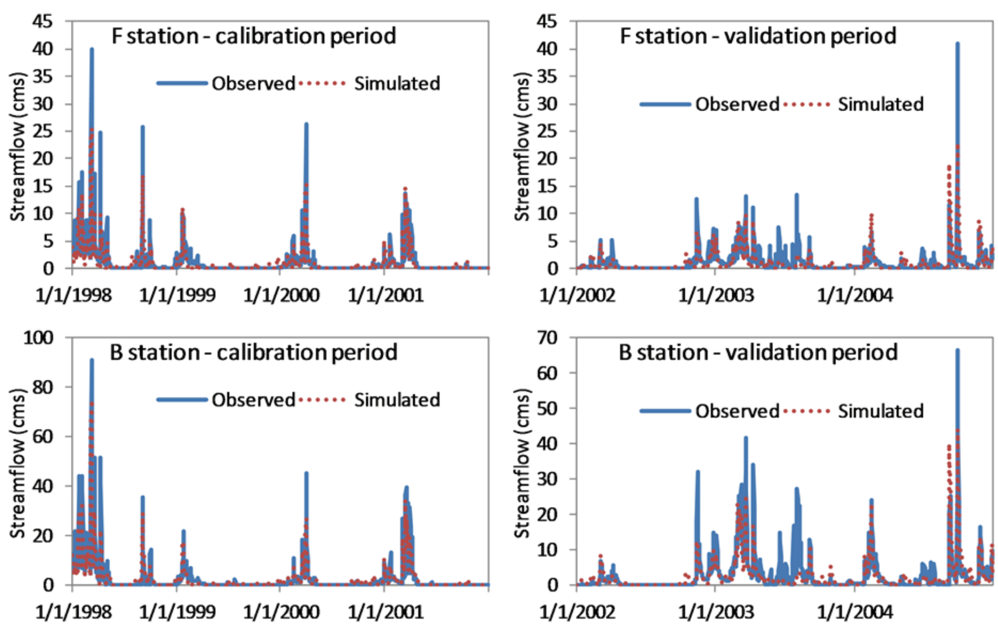


Figure 7. Observed and simulated daily streamflow in cubic meters per second (cms) at stations F and B for calibration and validation periods.

Conclusions

In this study, we developed a multi-core aware multi-objective evolutionary optimization tool, MAMEO, to improve the efficiency of calibrating SWAT. The test results demonstrated that, with a 16-core machine, MAMEO can save over 80% of the time (or several days) consumed by calibrating SWAT as compared with a sequential method executed on the same machine. Further analysis indicated that the capacity of the shared caches and I/O delay may influence the efficiency of MAMEO. The current version of MAMEO calls the SWAT executable program externally and does not have the flexibility to alter the directives within the SWAT model. Future work on re-engineering the structure of SWAT to optimize cache use and I/O operations deserves further attention to enhance the utilization of a multi-core platform to calibrate the SWAT model. Given its capability to efficiently produce multiple non-dominated parameter solutions for SWAT, MAMEO is expected to serve as a useful tool for SWAT users to explore the tradeoffs of multiple parameter solutions with respect to multiple performance metrics.

Acknowledgements

We would like to thank the three anonymous reviewers whose constructive comments substantially improved the quality of the manuscript. This work was funded by the DOE Great Lakes Bioenergy Research Center (DOE BER Office of Science DE-FC02-07ER64494, DOE BER Office of Science KP1601050, DOE EERE OBP 20469-19145) and NASA (NNH08ZDA001N and NNH12AU03I).

References

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 Optimization* 31(4): 635-672.

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

Bosch, D. D., J. M. Sheridan, R. R. Lowrance, R. K. Hubbard, T. C. Strickland, G. W. Feyereisen, and D. G. Sullivan. 2007. Little River Experimental Watershed database. *Water Resour. Res.* 43(9): W09470, doi: 10.1029/2006WR005844.

Confesor, R. B., and G. W. Whittaker. 2007. Automatic calibration of hydrologic models with multi-objective evolutionary algorithm and Pareto optimization. *J. American Water Res. Assoc.* 43(4): 981-989.

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

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

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

Moriasi, D. N., J. G. Arnold, M. W. Van Liew, R. L. Bingner, R. D. Harmel, and T. L. Veith. 2007. Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Trans. ASABE* 50(3): 885-900.

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

Neitsch, S. L., J. G. Arnold, J. R. Kiniry, R. Srinivasan, and J. R. Williams. 2005. *Soil and Water Assessment Tool (SWAT) Users Manual*. BRC Report 02-06. Temple, Tex.: Blackland Research Center, Texas Agricultural Experiment Station.

Nichols, J., S. Kang, W. M. Post, D. Wang, P. Bandaru, D. Manowitz, X. Zhang, and R. C. Izaurralde. 2011. HPC-EPIC for high-resolution simulations of environmental and sustainability assessment. *Computers and Electronics in Agric*. 79(2): 112-115.

Pacheco, P. 2011. *An Introduction to Parallel Programming*. 1st ed. Burlington, Mass.: Morgan Kaufmann.

Scott, L. R., T. Clark, and B. Bagheri. 2005. *Scientific Parallel Computing*. Princeton, N.J.: Princeton University Press.

Sexton, A., A. Sadeghi, X. Zhang, R. Srinivasan, and A. Shirmobammadi. 2010. Using NEXRAD and rain gauge precipitation data for hydrologic calibration of SWAT in a northeastern watershed. *Trans. ASABE* 53(5): 1501-1510.

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

Strauch, M., C. Bernhofer, S. Koide, M. Volk, C. Lorz, and F. Makeschin. 2012. Using precipitation data ensemble for uncertainty analysis in SWAT streamflow simulation. *J. Hydrol.* 414-415: 413-424.

Van Griensven, A., T. Meixner, S. Grunwald, T. Bishop, M. Di Luzio, and R. Srinivasan. 2006. A global sensitivity analysis tool for the parameters of multi-variable catchment models. *J. Hydrol.* 324(1-4): 10-23.

Van Liew, M. W., T. L. Veith, D. D. Bosch, and J. G. Arnold. 2007. Suitability of SWAT for the Conservation Effects Assessment Project: A comparison on USDA ARS watersheds. *J. Hydrol. Eng.* 12(2): 173-189.

Whittaker, G. 2004. Use of a Beowulf cluster for estimation of risk using SWAT. *Agron. J.* 96(5): 1495-1497.

Whittaker, G., R. Confesor, M. Di Luzio, and J. G. Arnold. 2010. Detection of overparameterization and overfitting in an automatic calibration of SWAT. *Trans. ASABE* 53(5): 1487-1499.

Williams, J. R. 1995. The EPIC model. In *Computer Models of Watershed Hydrology*, 909-1000. V. P. Singh, ed. Highlands Ranch, Colo.: Water Resources Publications.

Zhang, X, R. Srinivasan, K. Zhao, and M. Van Liew. 2009a. Evaluation of global optimization algorithms for parameter calibration of a computationally intensive hydrologic model. *Hydrol. Proc.* 23(3): 430-441.

Zhang, X., R. Srinivasan, and M. Van Liew. 2009b. Approximating SWAT model using artificial neural network and support vector machine. *J. American Water Resources Assoc.* 45(2): 460-474.

Zhang, X., R. Srinivasan, and D. Bosch. 2009c. Calibration and uncertainty analysis of the SWAT model using genetic algorithms and Bayesian model averaging. *J. Hydrol.* 374(3-4): 307-317.

Zhang, X., R. Srinivasan, and M. Van Liew. 2010. On the use of multi-algorithm, genetically adaptive, multi-objective method for multi-site calibration of the SWAT model. *Hydrol. Proc.* 24(8): 955-969.

Zhang, X., R. Srinivasan, J. G. Arnold, R. C. Izaurralde, and D. Bosch. 2011. Simultaneous calibration of surface flow and baseflow simulations: A revisit of the SWAT model calibration framework. *Hydrol. Proc.* 25(14): 2313-2320.

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

NOMENCLATURE

$\boldsymbol{x}$ = vector of hydrologic parameters in this study

$\boldsymbol{x}_i$ = ith parameter solution in the population: $\boldsymbol{x}_i = (\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, ..., \boldsymbol{x}_{iD})$

D = number of optimized parameters

$\boldsymbol{x}_{id}$ = dth dimension of the ith parameter solution

Ω = feasible space of parameters

$\boldsymbol{F}(\boldsymbol{x})$ = objective function vector that contains multiple individual objective functions that need to be optimized simultaneously: $\boldsymbol{F}(\boldsymbol{x}) = [f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), ..., f_m(\boldsymbol{x})]$

m = number of objective functions

P^* = Pareto optimal set

PF^* = Pareto front

T = maximum number of generations

t = current generation number

P_t = population of parameter solutions at generation t

N = number of parameter solutions in a population

P'_t = external archive at generation t that is used to store the parameter solutions with high fitness values

N' = external archive size