

# Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Parameter Calibration in Hydrological Simulation

Xinyu Zhang<sup>1</sup>, Yang Li<sup>1†</sup>, Genshen Chu<sup>1,2</sup>

<sup>1</sup>University of Science and Technology Beijing, Beijing 100083, China

<sup>2</sup>Engineering Research Center of Intelligent Supercomputing, Beijing, 100083

**Keywords:** Hydrologic simulation; Parameter calibration; Genetic algorithm; Particle swarm optimization

Citation: Zhang, X., Li, Y., Chu, G.: Comparison of Parallel Genetic Algorithm and Particle Swarm Optimization for Parameter Calibration in Hydrological Simulation. *Data Intelligence* 5(4), 904-922 (2023).

Received: September 16, 2023; Revised: October 20, 2023; Accepted: November 1, 2023

---

## ABSTRACT

Parameter calibration is an important part of hydrological simulation and affects the final simulation results. In this paper, we introduce heuristic optimization algorithms, genetic algorithm (GA) to cope with the complexity of the parameter calibration problem, and use particle swarm optimization algorithm (PSO) as a comparison. For large-scale hydrological simulations, we use a multilevel parallel parameter calibration framework to make full use of processor resources, and accelerate the process of solving high-dimensional parameter calibration. Further, we test and apply the experiments on domestic supercomputers. The results of parameter calibration with GA and PSO can basically reach the ideal value of 0.65 and above, with PSO achieving a speedup of 58.52 on TianHe-2 supercomputer. The experimental results indicate that using a parallel implementation on multicore CPUs makes high-dimensional parameter calibration in large-scale hydrological simulation possible. Moreover, our comparison of the two algorithms shows that the GA obtains better calibration results, and the PSO has a more pronounced acceleration effect.

---

## 1. INTRODUCTION

Hydrological models are widely used in many fields such as drought forecasting, flood protection and water resources management systems [1, 2, 3, 4]. The results of hydrological simulations depend to a large

---

<sup>†</sup> Corresponding author: Yang Li (Email: liyang@ustb.edu.cn; ORCID: 0000-0002-9331-2331).

extent on the merits of the model parameters. The process parameters in the hydrological model must be calibrated. Parameter calibration of hydrological model is an important process in which parameter adjustment is made so as to match (as closely as possible) the actual observed values. However, in complex practical applications, due to the high complexity of the hydrological process, the simulation of a large range of watersheds, and the need to calibrate more parameters, traditional optimization methods often suffer from a computational dimensionality curse. In addition, due to the long time consumption of the serial execution of the parameter calibration process, the excessively slow calibration speed seriously reduces the efficiency.

There are two types of parameter calibration methods: manual trial-and-error and automatic. In the era of less advanced computer technology, manual trial-and-error methods are widely used for parameter optimization in hydrologic models. The trial-and-error method requires manually adjusting parameters to match the simulated results with observed values, yet, the method not only takes a long time to debug but also depends on the human experience, which increases the uncertainty of the model [5, 6]. In addition, it is difficult to perform a complete sensitivity analysis due to a large number of self-interacting calibration parameters, which may prevent the suboptimal estimation of parameters [7]. In order to overcome the shortcomings of the manual trial-and-error method for parameter calibration problems, the exploration of the automatic calibration method has been launched. The automatic calibration involves with setting up objective functions, developing optimization algorithms and setting up termination conditions as calibration targets [8], where the optimization algorithms effectively avoid the subjectivity problem of the manual trial-and-error method [9]. Automatic calibration potentially leads to time savings and a more modeler-independent approach.

Traditional optimization algorithms, such as the gradient method and Newton method [10, 11], are susceptible to local convergence by initial values and often fail to obtain globally optimal solutions, resulting in strong instability of optimization results impossible to be promoted in practical applications [12, 13]. Since the 1980s, modern optimization algorithms, mainly including Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Shuffled Complex Evolution Algorithm (SCE-UA), Simulated Annealing Algorithms (SAA) and Artificial Neural Networks (ANN), have emerged and greatly promoted the development of parameter optimization of hydrological models [14, 15, 16, 17]. However, since the traditional calibration procedure is executed serially, which is usually computationally intensive, consumes many system resources, and causes a huge computational burden. In particular, in large-scale hydrological simulation, the running time may vary from several hours to several months [18], and the optimization efficiency problem becomes an important problem in the development of hydrological models. With the rapid development of high-performance computing, parallel computing has become an effective means to improve the efficiency of large-scale data processing, and combining optimization algorithms with parallelism can further improve feasibility [19, 20, 21, 22]. And yet, in terms of specific practical applications with optimization problems of high dimensionality and complexity, the curse of dimensionality causes slow convergence of the solution for parameters optimization and the individuals make the search easily fall into local optimal in the optimization process. In addition, the parallelization for calibrating hydrologic models requires load balancing and communication minimization.

In this paper, we use two heuristic algorithms to calibrate the parameters of hydrologic models on a large scale. There are three main contributions to our research work. First, we propose a multi-level parallel parameter calibration framework, where the inner parallel layer implements water cycle simulation and the outer parallel layer manipulates parameter iteration and optimization search by dividing subdomains. In our case, we use the GA and the PSO to calibrate parameters. Second, we propose a technique to parallelize both the GA and the PSO while minimizing the communication between the processes to make full use of computing resources with multicore CPUs. Finally, we compare the feasibility and parallel efficiency of the set of parameters generated by our GA-based and PSO-based parameter calibration on our domestic supercomputer. The results given in this paper clearly illustrate that parallel GA is suitable for applications requiring high calibration accuracy, parallel PSO is suitable for applications with fast calibration response.

The remainder of this paper is organized as follows. The next section discusses the related work. Then we present the multi-level parallel parameter calibration framework we applied in large-scale hydrologic simulation, followed using the GA and the PSO to calibrate parameters in hydrologic models. Finally, we present the parallel implementation of the GA and the PSO to allow parameter calibration on a large scale and compares the validation and performance of calibration results produced by the GA and the PSO, and summarize the conclusion.

## **2. RELATED WORK**

Parameter calibration is an important part that affects the accuracy and efficiency of large-scale hydrological simulation. The traditional parameter calibration method mainly adopts the manual experiment method, that is, a set of parameters is randomly selected from a given parameter range, and then the selected parameters are used to simulate [23]. Due to the large number and range of parameters, thousands of calculations are required during the calibration process. The obvious disadvantage of manual methods is that they consume much time and energy.

There are optimization algorithms used to calibrate hydrologic models [24, 25, 26, 27, 28], such as Gauss-Newton-type methods, Shuffled Complex Evolution of the University of Arizona (SCE-UA), Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and Artificial Neural Networks (ANN). Gauss-Newton-type algorithm as a local calibration method for the optimization of hydrological models, is usually more robust while keeping costs down but tends to get trapped in a local optimum when solving multi-extreme value problems. More efficient global optimization algorithms are proposed to consistently locate the global optimum of the hydrological models. The SCE-UA was used to calibrate in a complex watershed model [25] and the recommended values for the SCE-UA can be construed as guidelines for most applications. GA and PSO as optimization algorithms based on the natural intelligence of the biological world, both have no special restrictions on the linearity and continuity of the parameter calibration problem, and can provide a higher probability to obtain the global optimal solution when there are many local solutions, and have been widely used for parameter calibration of hydrological models [27, 29, 30, 31, 32, 33, 34]. Cheng et al. earlier successfully applied the GA to the parameter calibration of Xinanjiang model in China. Wang et

al. proposed a hybrid algorithm of GA and PSO to calibrate the SWAT model parameters, and applied it to the basin of the Zhangjiang River. Gao et al. extended the traditional PSO and applied it in calibrating the parameters of a water accounting model for the basin in Australia. The Bayesian approach [35] was applied to the Nash-Cascade model and Sacramento model. The authors used the Leaf River basin in Mississippi as an example, and verified that the Bayesian recursive estimation algorithm has good results in parameter optimization. However, these parameter optimization methods are executed serially, which can be time-consuming when applied to large-scale simulations. With the development of hydrological simulation combined with parallel technology, Kollet et al. [18] demonstrated the correctness and feasibility of using parallel computing for hydrological simulation. Yalew et al. [19] explored the use of parallel computing for hydrological simulation in the Soil and Water Assessment Tool (SWAT). A master-slave dynamic task assignment scheduling strategy was proposed to parallelize hydrological simulation by Li et al. [20, 21] Despite the introduction of parallel techniques and the corresponding improvement in the computational efficiency of hydrological models, these studies are aimed at the accelerated solution of the computational process of hydrological simulation and do not directly accelerate the process of parameter calibration. Rouholahnejad et al. [36] parallelized the calibration of the SWAT hydrological model. And, a parallel optimization algorithm Multi-core Parallel Artificial Bee Colony algorithm (MPABC) was proposed to improve the optimization precision and performance for parameters optimization of the Xinanjiang model [37]. A multicore parallel genetic algorithm (MCPGA) based on the fork-join parallel framework proposed is also applied to parameter calibration for the Xinanjiang model [38]. These studies have used multicore parallel technology to offer considerable benefits for parameter calibration, but most of their methods focus on improving optimization algorithms, and not many are based on parallel frameworks. We propose a multi-level parallel framework for large-scale hydrological simulation parameter calibration. By dividing the sub-communication domains, the framework is divided into multi-level parallel structures, and the introduction of the parallel framework has greatly improved the search efficiency and operating efficiency of the optimization algorithm compared to the serial.

### **3. MULTI-LEVEL PARALLEL PARAMETER CALIBRATION**

In order to meet the need for parameter optimization on a large scale and better improve computational efficiency, we introduce multi-level parallelism, with the inner layer, processes within each subdomain, using a custom message queue data structure and introducing parallelism in two dimensions of space-time for water cycle simulation; the outer layer, between subdomains, using Message passing interface (MPI) to divide the subdomain based on the master-slave model to parallelize the optimization space. The multilevel parallel parameter calibration framework is shown in Figure 1. MPI [39] is a parallel program interface with good performance. It has two most basic parallel program design modes: peer-to-peer mode and master-slave mode. In our work, the parameter calibration is based on the master-slave mode to realize parallelism. In this master-slave mode, we add a layer of internal master processes to the parallel framework, where the slave processes in each subdomain will first send the corresponding results to the internal master process, which will then perform the corresponding calculations and finally send the results to the external master process. Each internal master process is responsible for the communication of several adjacent subdomains,

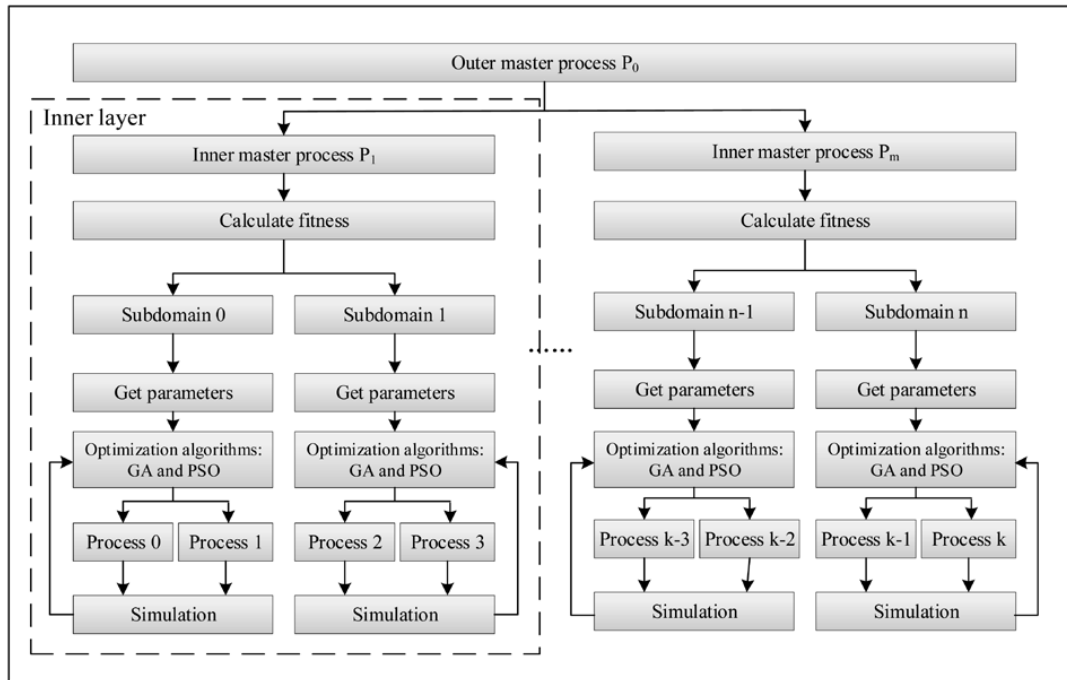


Figure 1. Multilevel parallel parameter calibration framework.

which effectively reduces the communication consumption of the outer master process and further accelerates the parameter calibration process.

We will focus on the parallel parameter calibration part in the outer layer. The way to divide the subdomain through the MPI is shown in Figure 2. The whole communication domain  $MPI\_COMM\_WORLD$  is divided into a new communication domain  $MPI\_NEW\_COMM$  by  $MPI\_Comm\_split$ , where process  $p_0$  is divided into a separate main domain and the rest of the processes form multiple subdomains. The multilevel parallel parameter calibration flowchart is shown in Figure 3. First, subdomains obtain  $N$  sets of parameters and perform the water cycle simulation. And then the simulation results are compared with corresponding measured values, and these results as well as the parameters are recorded in each iteration. When the subdomain has completed all the iterations, they send the above data to the inner master process in its own subdomain respectively, and the inner master process is responsible for calculating the optimal solutions and sending them to the external master process. The simulation results are further passed to the external master process for the final calibration. It is worth noting that we introduce GA and PSO to accelerate the optimization in the part of parameter calibration. For hydrological simulation, the measurement standard representing the hydrological simulation index can be used as a fitness function, such as the Nash-Sutcliffe efficiency (NSE) [40] or the determination coefficient ( $R^2$ ) [41]. The specific formula for calculating the NSE and  $R^2$  is defined as follows:

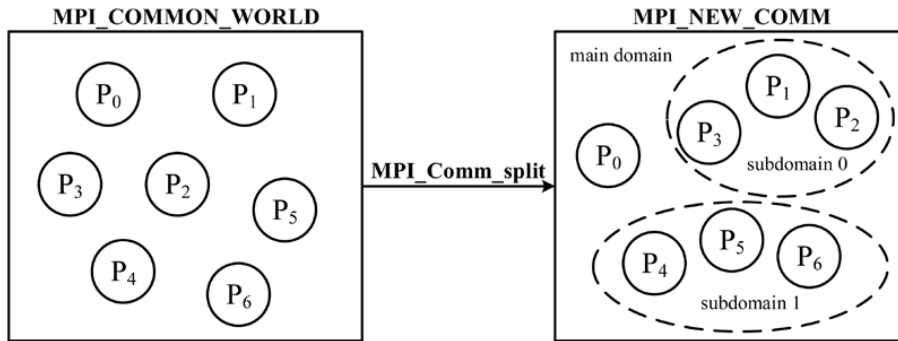


Figure 2. A domain divided into two subdomains.

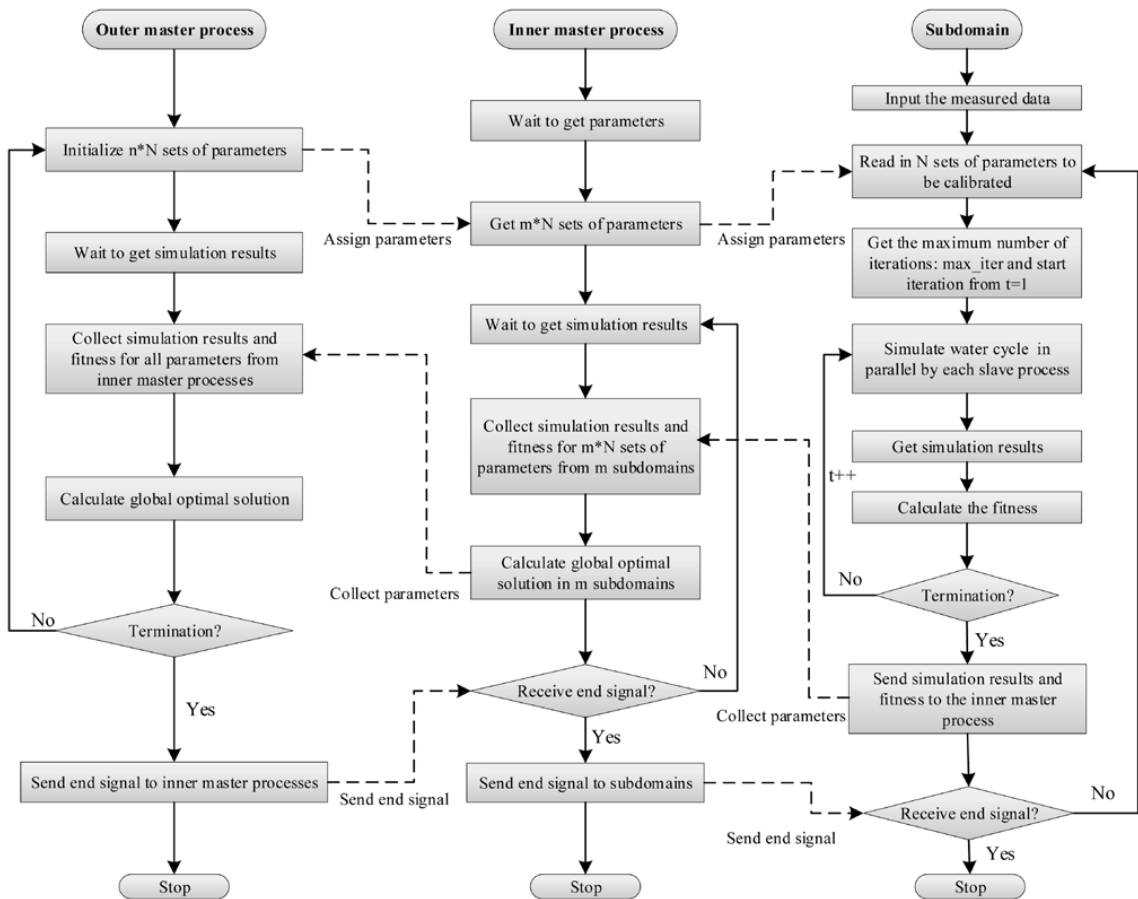


Figure 3. Multilevel parallel parameter calibration flowchart.

$$E_{ns} = 1 - \frac{\sum_{i=1}^n (Q_{obs,i} - Q_{sim,i})^2}{\sum_{i=1}^n (Q_{obs,i} - \bar{Q}_{sim,i})^2} \quad (1)$$

$$R^2 = \frac{\sum_{i=1}^n ((Q_{obs,i} - \bar{Q}_{obs})(Q_{sim,i} - \bar{Q}_{sim}))^2}{\sum_{i=1}^n (Q_{obs,i} - \bar{Q}_{obs})^2 \sum_{i=1}^n (Q_{sim,i} - \bar{Q}_{sim})^2} \quad (2)$$

where  $Q_{obs,i}$  is the observed value of the  $i$ th sub-basin and  $Q_{sim,i}$  is the simulated value of the  $i$ th sub-basin. The NSE, also known as the simulation efficiency coefficient, takes a value range of  $-\infty$  to 1. If the NSE close to 1, the simulation quality is good; close to 0, the overall results are credible simulation process error is large; much less than 0, the model is not credible. The coefficient of determination  $R^2$  takes the value of 0.0 to 1.0, and the ideal value is 0.8 to 1.0. The  $R^2$  closer to 1 means that the difference between the simulated runoff and the measured runoff is smaller.

#### 4. PALLEL GENETIC ALGORITHM (PGA)

The GA is an adaptive heuristic search algorithm based on the evolutionary process of natural selection and genetics [42]. The GA simulates the evolution process of the natural population, and selects the optimal individual which adapts to the environment by eliminating backward genes, and the solution of the optimization problem can be regarded as a set of chromosomes. Because of its high versatility, strong robustness and ability to deal with non-analytical objective functions and constraints, GA has been used for the calibration of conceptual runoff models and optimization of cascade stilling [26, 27, 43]. As the scale of the optimization problem increases and the search space becomes more complex, with the GA inherent parallelism, the parallel genetic algorithm (PGA) becomes the most significant way to solve these complicated problems efficiently and effectively [44]. In this work, we use master-slave PGA [45] to simulate the evolution of a population with multiple sets of parameters in hydrological simulation adapting to the objective function. The flowchart of our parallel GA is displayed in Figure 4.

In our implementation, a set of parameters of the sub-basins is a set of chromosomes, and a specific parameter value on the chromosome represents a gene. The essence of GA is to obtain the optimal solution that satisfies the condition by changing the value of the gene. The flowchart of the GA includes chromosome selection, crossover and mutation. The initial population with multiple sets of parameters is randomly generated within the set parameter value range. The chromosome selection is mainly based on the roulette wheel selection [46] to ensure randomness and the selection criterion is the fitness of the solution that refers to the objective function we ultimately need to solve. The fitness evaluation of individuals is assigned to different processors. In our hydrological simulation, we take the  $R^2$  coefficient as the fitness. The serial operation of selection consists of the following steps. First, the  $R^2$  coefficient of  $N$  sets of parameters for water cycle simulation is calculated and the probability of each set of parameters being inherited by the next generation population can be calculated by:

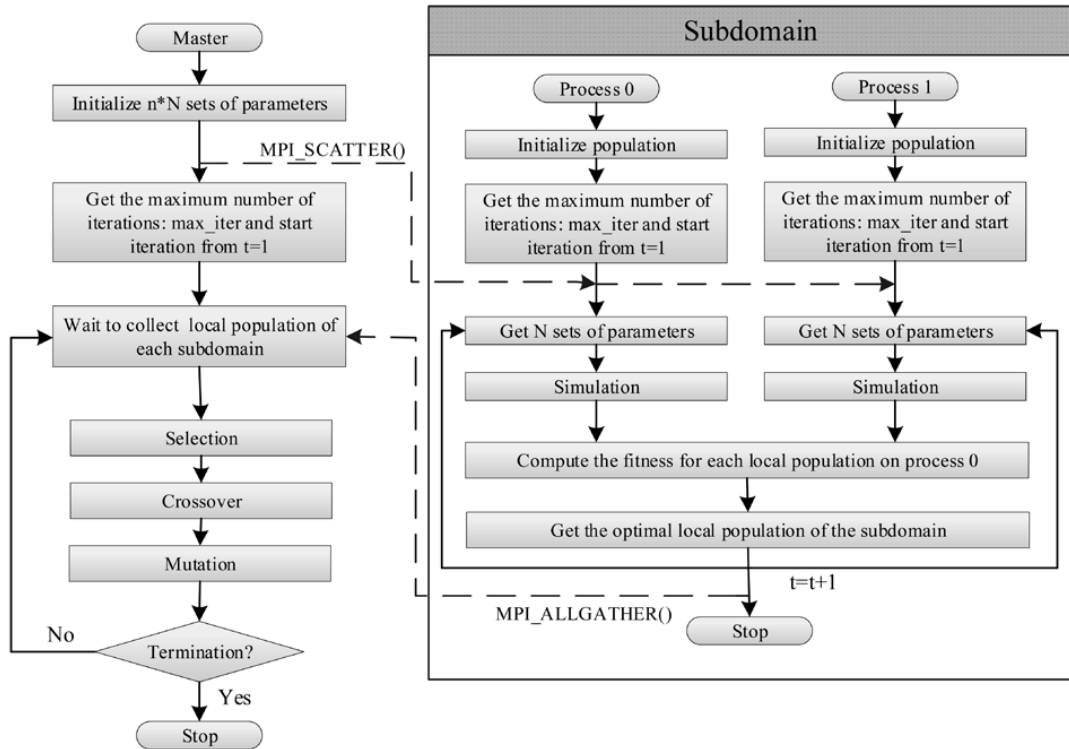


Figure 4. Flowchart of our parallel genetic algorithm.

$$P(X_i) = \frac{f(x_i)}{\sum_{j=1}^N f(x_j)} \quad (3)$$

where  $P(X_i)$  is the inheritance probability of the  $i$ th chromosome, and  $f(x_i)$  is the  $R^2$  coefficient of the  $i$ th chromosome. The inheritance probability of each chromosome is then accumulated. Second, a uniformly distributed value is randomly generated in  $[0,1]$ , if the value is less than the cumulative probability of the first chromosome, the first chromosome can be selected, and if the value is between the cumulative probability of two chromosomes, the latter chromosome is selected.  $N$  chromosomes, which are  $N$  sets of the parameters, can be selected by repeating the above steps  $N$  times.

When the parents are selected, children can be created from parents using the single-point crossover. Significantly, if the number of chromosomes is odd, the last chromosome is directly retained. The children solutions need to mutate to guarantee the diversity of the population and avoid the local convergence caused by the crossover. According to the number of parameters in the current sub-basins, mutation points are randomly set for the chromosomes, and the value at the mutation point should be set within the value range given in the parameter configuration file. Finally, the optimal target value is constantly updated and



the evolution cycle continues until the termination condition has been met. The termination condition in our implementation is defined as the maximum number of iterations of simulation or final convergence accuracy.

## 5. PARTICLE SWARM OPTIMIZATION (PSO)

The PSO [47] is a population-based stochastic optimization method that was proposed by Eberhart and Kennedy in 1995. It simulates the swarm behavior which looks for food in a cooperative way of insects, herds, birds and fish. For example, when a group of birds is looking for food, each bird inside will be abstracted into a particle with position and speed, and each particle will change its possible future activity trend and trajectory according to its own experience as well as the experience of other members. Compared with GA, the PSO model has the advantages of fewer parameters, simple principles and easy implementation. Similarly to the GA, the PSO has been used for parameter calibration in some hydrological models [33, 48, 49]. The algorithm is initialized to a group of random particles, that is random solutions, and then the optimal solution is obtained iteratively. At every step of the iteration, the velocity of each particle is updated according to the current individual best position found by itself and the current global best position shared by the whole particle swarm [50]. If the swarm consists of  $N$  particles and the maximum iteration counter is  $max\_iter$ , the velocity and position of a single particle are updated at  $t$  iteration by the following formula:

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + c_1 \mathbf{u}_1^t \cdot (\mathbf{pb}_i^t - \mathbf{x}_i^t) + c_2 \mathbf{u}_2^t \cdot (\mathbf{gb}_i^t - \mathbf{x}_i^t) \quad i \in \{1, 2, \dots, N\} \quad (4)$$

$$\mathbf{x}_{t+1} = \mathbf{x}_t + \mathbf{v}_{t+1} \quad 1 \leq t \leq max\_iter \quad (5)$$

where vectors  $\mathbf{v}$  and  $\mathbf{x}$  are the velocity and the position of the particle separately;  $\mathbf{pb}$  is the best position previously occupied by the particle;  $\mathbf{gb}$  is the best position previously occupied by any particle of the global swarm;  $\mathbf{u}_1$  and  $\mathbf{u}_2$  are vectors of random values between 0 and 1. The velocity and the position of each particle in the global space are updated independently by the above two equations. The flowchart of our parallel PSO is shown in Figure 5.

In our work, the position vector of the particle corresponds to a set of parameters in the water cycle simulation process. We will take a simple model of three sub-basins, of which the first and the second are upstream sub-basins and the last one is the corresponding downstream sub-basin as an example to explain the specific meaning of particle position and velocity in the simulation. The relationship between input  $x$  and output  $y$  in each sub-basin is defined as:

$$\begin{aligned} y_1 &= k_1 * x_1 + b_1 \\ y_2 &= k_2 * x_2 + b_2 \\ y_3 &= k_3 * x_3 + b_3 \end{aligned} \quad (6)$$

and the vector  $\mathbf{x}$  which is each particle's position representing a set of parameters for all sub-basins is defined as:

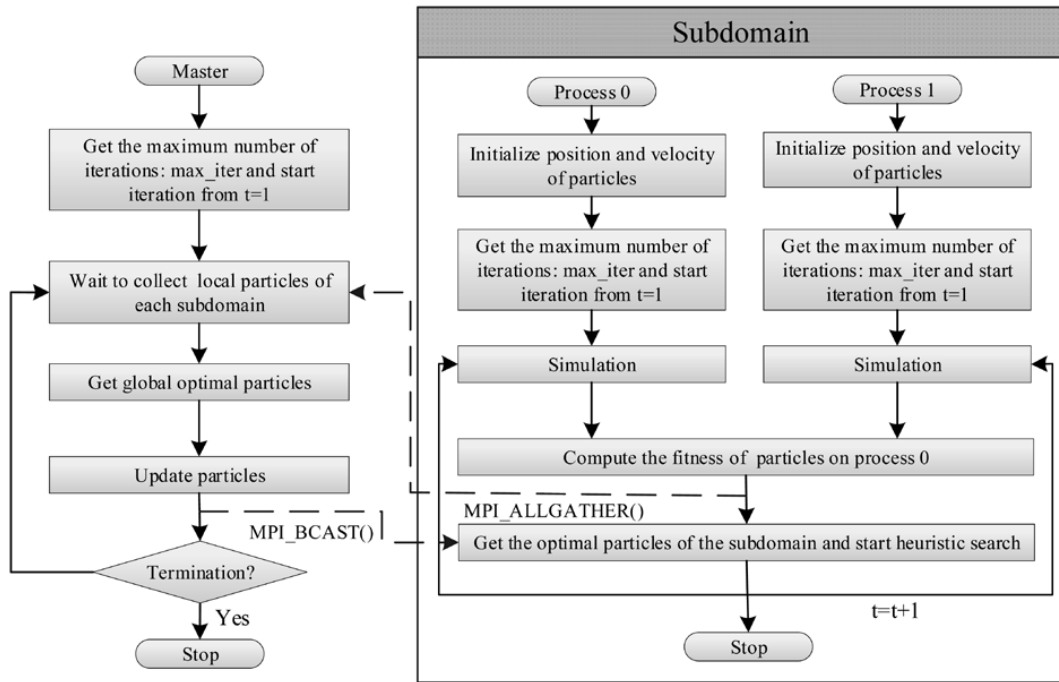


Figure 5. Flowchart of our parallel particle swarm algorithm.

$$\mathbf{x} = [k_1, b_1, k_2, b_2, k_3, b_3]^T \quad (7)$$

and the  $\mathbf{v}$  which is each particle's velocity representing the variation of each parameter is also a vector corresponding to the position, that is,

$$\mathbf{v} = [v_1, v_2, v_3, v_4, v_5, v_6]^T \quad (8)$$

If the component direction of  $\mathbf{v}$  is positive, it means that every component of the vector  $\mathbf{x}$  will increase by the corresponding value, and if the component direction is negative, it will decrease. As for the best solution, the local optimal solution of each particle refers to the parameter values corresponding to the optimal  $R^2$  coefficient that appears for each particle as of the current time step, and the global optimal solution corresponds to the optimal coefficient appearing in all particles as of the current time step.

## 6. EXPERIMENTS AND RESULTS

In the above sections, we described how to calibrate the parameters in water cycle simulation and find out the optimal set of parameters. Although a set of optimal parameters can be selected through specific algorithms and methods to make the hydrological simulation results closest to the actual situation and

achieve the optimal effect, the demand of calculation for parameters of each sub-basin will increase geometrically in large-scale simulation, resulting in too long time for real-time applications. We need to meet the calibration requirements in high spatial dimension and long time span in this large-scale case, so we developed parallel versions of our GA and PSO. In our parallel implementation, we adopt a master-slave implementation and split the domain owned by the MPI processor to make full use of the advantages of multicore CPUs. The optimization space of the whole set of parameters is allocated to each subdomain for calculation, and the algorithm is introduced into each subdomain to speed up the iteration. Our approach maintains a good level of cooperation among subdomains, performs parameter iteration in parallel, accelerates parameter calibration, and makes full use of computing resources.

In our experiments, we test parallel GA and PSO using drainage network data, which consists of hydrological models information, latitude and longitude, area, precipitation values and observed values of Gan River (23 sub-basins) which is located in China. We run the parallel hydrologic simulation program with calibration optimizations on our national supercomputer, TianHe-2, which ranked tenth in top500 list in November 2022. We simulate one year of rainfall-runoff and routing process using the Wetspa runoff model and Muskingum routing model in our parallel hydrologic simulation program. The initial conditions are shown in Table 1. We first calibrate the hydrological data for the first half of 2018, and then use this model to simulate for the second half of the year. As previously stated, the fitting accuracy we used is measured by index  $R^2$ , and the closer the value is to 1, the better the calibration effects. The trend of  $R^2$  based on the GA and PSO within 100 iterations during the calibration period is shown in Figure 6. It can be seen that the calibration results converge to 0.74 and 0.65 based on GA and PSO, respectively, as the number of iterations increases. The convergence values of the algorithms during the simulation period are shown in Table 2, where the final convergence result of GA is higher than that of PSO, and consistent with results in the calibration period. In addition, the fitting accuracy of the calibration results associated with GA exceeded 0.8 to meet the desired expectations after 1000 iterations. It can be concluded that our method works. Moreover, it can be analyzed that the iteration of PSO depends on its own optimal value as well as the global optimal value, so it is easy to make parameters fall into the local optimal value. Since the GA operates with genetic mutation when performing parameter calibration, it ensures that the population is diverse and therefore looks beyond the global optimum to find a wider range of solutions.

**Table 1.** Algorithm parameter values.

Parameters	Values
Number of iterations	100,500 and 1000
Calibration accuracy	0.8
Crossover rate (GA only)	0.88
Mutation rate (GA only)	0.1
$c_1$ (PSO only)	2
$c_2$ (PSO only)	2

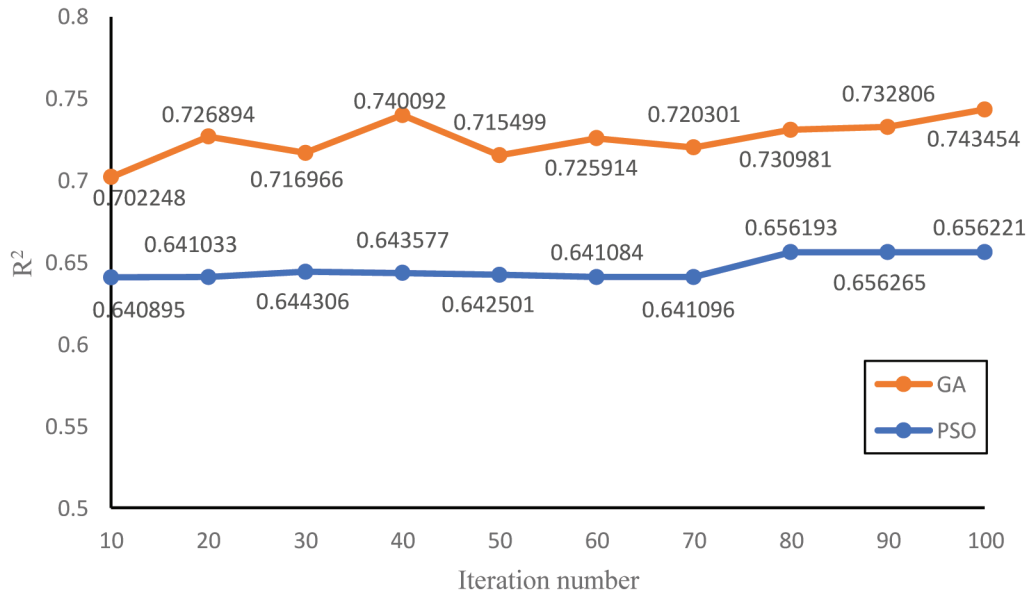


Figure 6. The calibration result of GA and PSO at each iteration.

Table 2. Comparison of convergence values of GA and PSO.

Number of iterations	Algorithms	
	GA	PSO
100	0.761534	0.639657
500	0.791308	0.639676
1000	0.831248	0.639826

We run the parallel calibration process with 8192 parameters and test the performance of the GA and the PSO with 32, 64, 128, 256, 1024, and 2048 processes on Tianhe-2 for Gan River dataset. In addition, we recorded the average simulation time of five repeat simulations for each case. The speedup and parallel efficiency of our parallel GA and PSO are plotted in Figure 7 and Figure 8. We can find that the speedup of GA and PSO increases as the number of processes increases, but the speedup of the GA slows down at 256 processes, while the PSO still increases at a speedup close to the ideal. As the parameter update of the GA is more complex, the master process takes up the core work, such as parameter selection, crossover and mutation, while the slave processes are mainly responsible for the calculation of the water cycle simulation, so the computation of the master process cannot be effectively reduced in the process of adding slave processes. In contrast, the parameter update of the PSO is simpler, and its speedup increases significantly as the number of processes increases. However, due to the consumption of process communication, the more the number of processors, the more frequent the communication will be, which in turn leads to a decrease in the parallel efficiency of the processor as the number of processes increases.

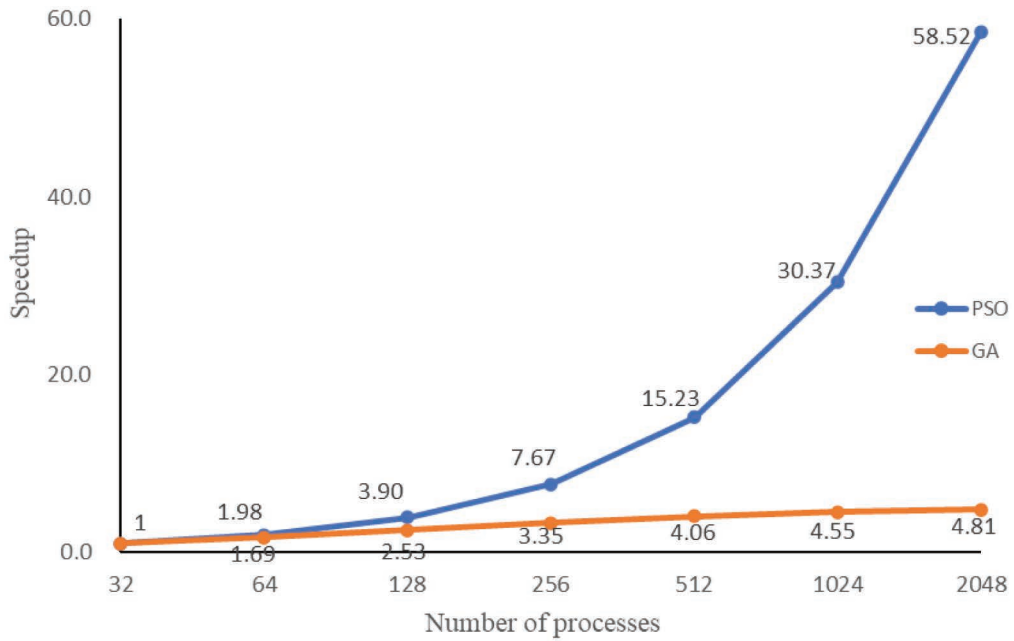


Figure 7. The speedup of our parallel methods with GA and PSO for different work sizes.

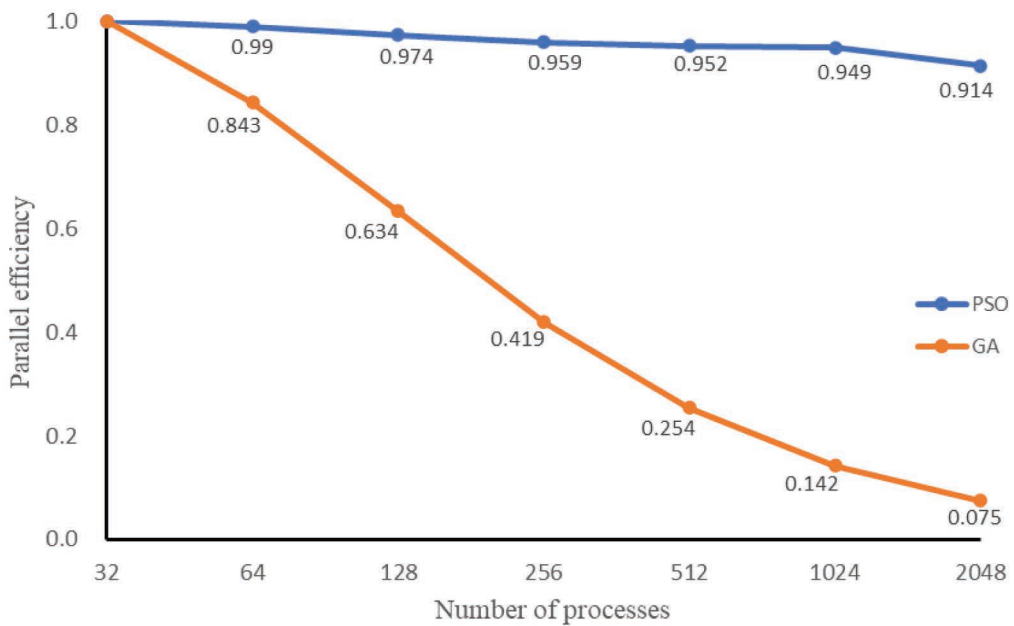
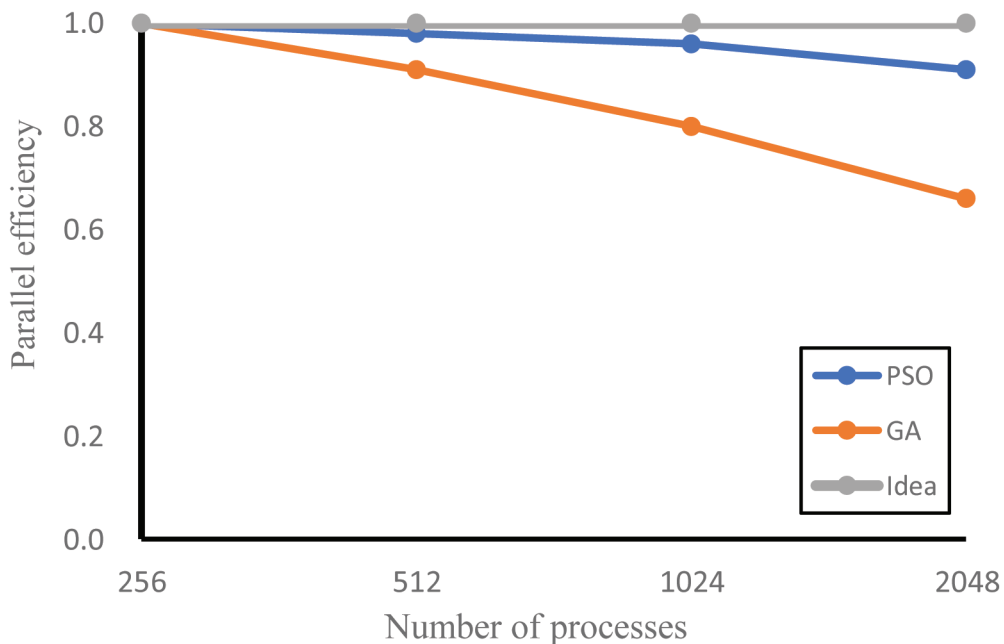


Figure 8. The parallel efficiency of our parallel methods with GA and PSO for different work sizes.

In order to further test the parallel performance of parameter calibration for large scale sub-basins, we will discuss our study on the scalability of our simulation on Tianhe-2. We use calculation examples of 1744 sub-basins to carry out a water cycle simulation test with a duration of 10 days and expand the number of processes to 2048 and take the performance of 256 cores as a baseline. Figure 9 shows the parallel efficiency of the case with parallel GA and PSO respectively for scalability when the number of processes varies from 256 to 2048. It can be seen that whether it is parallel GA or PSO, our parallel optimization work for large-scale calibration simulation has been reflected on our domestic supercomputer, and we can find that there is less performance loss as the scale increase for PSO. The result indicates that our parallel optimization work has the ability to expand hydrological simulation to a larger scale in massive parallelism on our domestic supercomputer.



**Figure 9.** Results of scalability performance tests with GA and PSO.

From the above test results and summary, it can be concluded that the parallel genetic algorithm and the parallel particle swarm algorithm are suitable for different hydrological simulation applications due to their respective characteristics: the parallel genetic algorithm is used for accurate hydrological models or long-term simulations, and parallel particle swarm algorithm is used for short-term simulations for real-time predictions. Moreover, the parallel optimization work we carried out on the domestic supercomputer accelerated the parameter calibration process, which in turn made the simulation have higher parallel performance.

## 7. CONCLUSION

This paper presents a parameters optimization solution for large-scale hydrological simulation which considers the multiplicity of parameters and the complexity of sub-basins. We used two optimization algorithms, the GA and the PSO, to attack the complexity of high dimensional parameters and produce solutions in a relatively short computation time. The computing time is further reduced through the parallelization of optimization algorithms. After obtaining convergence results essentially around 0.65 and the parallel efficiency of 91% on 2048 cores, we conclude that, it is possible to achieve high dimensional parameters calibration in large-scale hydrological simulation by using a parallel implementation on our domestic supercomputer, that is the optimal parameters can be obtained in a relatively short computation time while ensuring correctness of simulation results, and also has an acceleration effect in larger scale simulation. In addition, our comparison of the two algorithms shows that the final result of convergence of the GA is better than the PSO, and that the acceleration effect of the PSO is more obvious than the GA. It is necessary to combine the characteristics of the two algorithms in the specific use of parameters.

## AUTHOR CONTRIBUTIONS

This work was a collaboration between all of the authors. X.Y. Zhang (xy\_zhangzhang@foxmail.com) conducted the experiment, wrote and revised the manuscript. Y. Li (liyang@ustb.edu.cn) proposed the idea of research and provided insightful revision on the manuscript. G.S. Chu (genshenchu@gmail.com) designed the experiment and prepared the data set. All authors have made valuable and meaningful contributions to the manuscript.

## ACKNOWLEDGMENTS

The research is supported by National Key R&D Program of China No. 2021YFB0300202 & 2021YFB0300200. and Interdisciplinary Research Project for Young Teachers of USTB (Fundamental Research Funds for the Central Universities) No. FRF-IDRY-20-036.

## REFERENCES

- [1] Hauswirth, S. M., Bierkens, M. F. P., Beijk, V., et al.: The suitability of a seasonal ensemble hybrid framework including data-driven approaches for hydrological forecasting. *Hydrology and Earth System Sciences* 27, 501–517 (2023). <https://doi.org/10.5194/hess-27-501-2023>
- [2] Kratzert, F., Klotz, D., Brenner, C., et al.: Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks. *Hydrology and Earth System Sciences* 22(11), 6005–6022 (2018). <https://doi.org/10.5194/hess-22-6005-2018>
- [3] Zhang, L., Yang, X.: Applying a Multi-Model Ensemble Method for Long-Term Runoff Prediction under Climate Change Scenarios for the Yellow River Basin, China. *Water* 10(3), 301 (2018). <https://doi.org/10.3390/w10030301>

- [4] Nayeb Yazdi, M., Ketabchy, M., Sample, D. J., et al.: An evaluation of HSPF and SWMM for simulating stream-flow regimes in an urban watershed. *Environmental Modelling & Software* 118, 211–225 (2019). <https://doi.org/10.1016/j.envsoft.2019.05.008>
- [5] Yapo, P. O., Gupta, H. V., Sorooshian, S., et al.: Multi-objective global optimization for hydrologic models. *Journal of Hydrology* 204(1–4), 83–97 (1998). [https://doi.org/10.1016/s0022-1694\(97\)00107-8](https://doi.org/10.1016/s0022-1694(97)00107-8)
- [6] Vrugt, J. A., Gupta, H. V., Bastidas, L. A., et al.: Effective and efficient algorithm for multiobjective optimization of hydrologic models. *Water Resources Research* 39(8), (2003). <https://doi.org/10.1029/2002wr001746>
- [7] Perin, R., Trigatti, M., Nicolini, M., et al.: Automated calibration of the EPA-SWMM model for a small suburban catchment using PEST: a case study. *Environmental Monitoring and Assessment* 192(6), (2020). <https://doi.org/10.1007/s10661-020-08338-7>
- [8] Shahed Behrouz, M., Zhu, Z., Matott, L. S., et al.: A new tool for automatic calibration of the Storm Water Management Model (SWMM). *Journal of Hydrology* 581, 124436 (2020). <https://doi.org/10.1016/j.jhydrol.2019.124436>
- [9] Nandi, S., Janga Reddy, M.: Comparative performance evaluation of self-adaptive differential evolution with GA, SCE and DE algorithms for the automatic calibration of a computationally intensive distributed hydrological model. *H2Open Journal* (2020). <https://doi.org/10.2166/h2oj.2020.030>
- [10] Gan, T. Y., Biftu, G. F.: Automatic Calibration of Conceptual Rainfall-Runoff Models: Optimization Algorithms, Catchment Conditions, and Model Structure. *Water Resources Research* 32(12), 3513–3524 (1996). <https://doi.org/10.1029/95wr02195>
- [11] Boyle, D. P., Gupta, H. V., Sorooshian, S.: Toward improved calibration of hydrologic models: Combining the strengths of manual and automatic methods. *Water Resources Research* 36(12), 3663–3674 (2000). <https://doi.org/10.1029/2000wr900207>
- [12] White, L. W., Vieux, B., Armand, D., et al.: Estimation of optimal parameters for a surface hydrology model. *Advances in Water Resources* 26(3), 337–348 (2003). [https://doi.org/10.1016/s0309-1708\(02\)00189-6](https://doi.org/10.1016/s0309-1708(02)00189-6)
- [13] Seibert, J.: Multi-criteria calibration of a conceptual runoff model using a genetic algorithm. *Hydrology and Earth System Sciences* 4(2), 215–224 (2000). <https://doi.org/10.5194/hess-4-215-2000>
- [14] Jahandideh-Tehrani, M., Bozorg-Haddad, O., Loáiciga, H. A.: Application of particle swarm optimization to water management: an introduction and overview. *Environmental Monitoring and Assessment* 192(5), (2020). <https://doi.org/10.1007/s10661-020-8228-z>
- [15] Rahnamay Naeini, M., Analui, B., Gupta, H., et al.: ‘Three decades of the Shuffled Complex Evolution (SCE-UA) optimization algorithm: Review and applications’, *Scientia Iranica* 26(Issue 4: Special Issue Dedicated to Professor Abolhassan Vafai), 2015-2031 (2019). <https://doi.org/10.24200/sci.2019.21500>
- [16] Feng, Z., Niu, W.: Hybrid artificial neural network and cooperation search algorithm for nonlinear river flow time series forecasting in humid and semi-humid regions. *Knowledge-Based Systems* 211, 106580 (2021). <https://doi.org/10.1016/j.knosys.2020.106580>
- [17] Efstratiadis, A., Koutsoyiannis, D.: One decade of multi-objective calibration approaches in hydrological modelling: a review. *Hydrological Sciences Journal* 55(1), 58–78 (2010). <https://doi.org/10.1080/02626660903526292>
- [18] Kollet, S. J., Maxwell, R. M., Woodward, C. S., et al.: Proof of concept of regional scale hydrologic simulations at hydrologic resolution utilizing massively parallel computer resources. *Water Resources Research* 46(4), (2010). <https://doi.org/10.1029/2009wr008730>
- [19] Yalew, S. G., Van Griensven, A., Kokozkiewicz, L.: Parallel computing of a large scale spatially distributed model using the Soil and Water Assessment Tool (SWAT). *International Environmental Modelling and Software Society (IEMSs) 2010 International Congress on Environmental Modelling and Software Modelling for Environment’s Sake* (2010)



- [20] Li, T., Wang, G., Chen, J., et al.: Dynamic parallelization of hydrological model simulations. *Environmental Modelling & Software* 26(12), 1736–1746 (2011). <https://doi.org/10.1016/j.envsoft.2011.07.015>
- [21] Li, Y., Jiang, D. L.: Dynamic Task-Scheduling Based Parallel Processing on Watershed Distributed Eco-Hydrological Model. *Applied Mechanics and Materials* 556-562, 3492–3495 (2014). <https://doi.org/10.4028/www.scientific.net/amm.556-562.3492>
- [22] Asgari, M., Yang, W., Lindsay, J., et al.: A review of parallel computing applications in calibrating watershed hydrologic models. *Environmental Modelling & Software* 151, 105370 (2022). <https://doi.org/10.1016/j.envsoft.2022.105370>
- [23] Kim, S. M., Benham, B. L., Brannan, K. M., et al.: Comparison of hydrologic calibration of HSPF using automatic and manual methods. *Water Resources Research* 43(1), (2007). <https://doi.org/10.1029/2006wr004883>
- [24] Moradkhani, H., Sorooshian, S.: General Review of Rainfall-Runoff Modeling: Model Calibration, Data Assimilation, and Uncertainty Analysis. *Water Science and Technology Library*, pp. 1–24 (2009). [https://doi.org/10.1007/978-3-540-77843-1\\_1](https://doi.org/10.1007/978-3-540-77843-1_1)
- [25] Duan, Q., Sorooshian, S., Gupta, V. K.: Optimal use of the SCE-UA global optimization method for calibrating watershed models. *Journal of Hydrology* 158(3–4), 265–284 (1994). [https://doi.org/10.1016/0022-1694\(94\)90057-4](https://doi.org/10.1016/0022-1694(94)90057-4)
- [26] Bakhtyar, R., Barry, D. A.: Optimization of cascade stilling basins using GA and PSO approaches. *Journal of Hydroinformatics* 11(2), 119–132 (2009). <https://doi.org/10.2166/hydro.2009.046>
- [27] Wang, Q. J.: The Genetic Algorithm and Its Application to Calibrating Conceptual Rainfall-Runoff Models. *Water Resources Research* 27(9), 2467–2471 (1991). <https://doi.org/10.1029/91wr01305>
- [28] Qin, Y., Kavetski, D., Kuczera, G.: A Robust Gauss – Newton Algorithm for the Optimization of Hydrological Models: From Standard Gauss – Newton to Robust Gauss – Newton. *Water Resources Research* 54(11), 9655–9683 (2018). <https://doi.org/10.1029/2017wr022488>
- [29] Zhang, X., Srinivasan, R., Zhao, K., et al.: Evaluation of global optimization algorithms for parameter calibration of a computationally intensive hydrologic model. *Hydrological Processes* 23(3), 430–441 (2009). <https://doi.org/10.1002/hyp.7152>
- [30] Cheng, C. T., Ou, C. P., Chau, K. W.: Combining a fuzzy optimal model with a genetic algorithm to solve multi-objective rainfall-runoff model calibration. *Journal of Hydrology* 268(1–4), 72–86 (2002). [https://doi.org/10.1016/s0022-1694\(02\)00122-1](https://doi.org/10.1016/s0022-1694(02)00122-1)
- [31] Wang, Y., Zhu, S., Yuan, L., et al.: An automatic parameter calibration method for the SWAT model in runoff simulation. *River Research and Applications* 36(7), 1321–1333 (2020). <https://doi.org/10.1002/rra.3655>
- [32] Ahli, H., Merabtene, T., Seddique, M.: Optimization of a Conceptual Rainfall-Runoff Model using Evolutionary Computing methods. 2021 14th International Conference on Developments in ESystems Engineering (DeSE) (2021). <https://doi.org/10.1109/dese54285.2021.9719369>
- [33] Gao, L., Kirby, M., Ahmad, M.-D., et al.: Automatic calibration of a whole-of-basin water accounting model using a comprehensive learning particle swarm optimiser. *Journal of Hydrology* 581, 124281 (2020). <https://doi.org/10.1016/j.jhydrol.2019.124281>
- [34] Napiorkowski, J. J., Piotrowski, A. P., Karamuz, E., et al.: Calibration of conceptual rainfall-runoff models by selected differential evolution and particle swarm optimization variants. *Acta Geophysica* (2022). <https://doi.org/10.1007/s11600-022-00988-0>
- [35] Thiemann, M., Trosset, M., Gupta, H., et al.: Bayesian recursive parameter estimation for hydrologic models. *Water Resources Research* 37(10), 2521–2535 (2001). <https://doi.org/10.1029/2000wr900405>
- [36] Rouholahnejad, E., Abbaspour, K. C., Vejdani, M., et al.: A parallelization framework for calibration of hydrological models. *Environmental Modelling & Software* 31, 28–36 (2012). <https://doi.org/10.1016/j.envsoft.2011.12.001>

- [37] Huo, J., Liu, L., Zhang, Y.: An improved multi-cores parallel artificial Bee colony optimization algorithm for parameters calibration of hydrological model. *Future Generation Computer Systems* 81, 492–504 (2018). <https://doi.org/10.1016/j.future.2017.07.020>
- [38] Liao, S., Sun, Q., Cheng, C., et al.: Multicore Parallel Genetic Algorithm with Tabu Strategy for Rainfall-Runoff Model Calibration. *Journal of Hydrologic Engineering* 22(8), (2017). [https://doi.org/10.1061/\(asce\)he.1943-5584.0001542](https://doi.org/10.1061/(asce)he.1943-5584.0001542)
- [39] Gropp, W., Lusk, E., Doss, N., et al.: A high-performance, portable implementation of the MPI message passing interface standard. *Parallel Computing* 22(6), 789–828 (1996). [https://doi.org/10.1016/0167-8191\(96\)00024-5](https://doi.org/10.1016/0167-8191(96)00024-5)
- [40] McCuen, R. H., Knight, Z., Cutter, A. G.: Evaluation of the Nash–Sutcliffe Efficiency Index. *Journal of Hydrologic Engineering* 11(6), 597–602 (2006). [https://doi.org/10.1061/\(asce\)1084-0699\(2006\)11:6\(597\)](https://doi.org/10.1061/(asce)1084-0699(2006)11:6(597))
- [41] Legates, D. R., McCabe, G. J.: Evaluating the use of “goodness-of-fit” Measures in hydrologic and hydroclimatic model validation. *Water Resources Research* 35(1), 233–241 (1999). <https://doi.org/10.1029/1998wr900018>
- [42] Holland, J. H.: *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control and artificial intelligence*. Ann Arbor, Mich (1975)
- [43] Ma, Y., Sheng, W., Li, J., et al.: Parameter Calibration of SWMM Model Based on GA. *China Rural Water and Hydropower* 7, 46–49, 53 (2020). <https://doi.org/10.3969/j.issn.1007-2284.2020.07.009>
- [44] Xue, S., Guo, S., Bai, D.: The Analysis and Research of Parallel Genetic Algorithm. 2008 4th International Conference on Wireless Communications, Networking and Mobile Computing 1–4 (2008). <https://doi.org/10.1109/wicom.2008.1275>
- [45] Tavares, L. G., Lopes, H. S., Lima, C. R. E.: A study of topology in insular Parallel Genetic Algorithms. 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC) (2009). <https://doi.org/10.1109/nabic.2009.5393437>
- [46] Blickle, T., Thiele, L.: A Comparison of Selection Schemes Used in Evolutionary Algorithms. *Evolutionary Computation* 4(4), 361–394 (1996). <https://doi.org/10.1162/evco.1996.4.4.361>
- [47] Kennedy, J., Eberhart, R.: Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks* (2019). <https://doi.org/10.1109/icnn.1995.488968>
- [48] Tang, X., Zhang, J., Wang, G., et al.: Uncertainty Analysis of SWAT Modeling in the Lancang River Basin Using Four Different Algorithms. *Water* 13(3), 341 (2021). <https://doi.org/10.3390/w13030341>
- [49] Qqunta, J. G., Astorayme, M. A., Gutiérrez, R. R., et al.: Assessment, Evaluation, and Code Development of the Particle Swarm Optimisation (PSO) Method for an Automatic Calibration of the TOPMODEL. *World Environmental and Water Resources Congress 2022* (2022). <https://doi.org/10.1061/9780784484258.114>
- [50] Clerc, M.: Combinatorial Problems. *Particle Swarm Optimization*, 201–210 (2006). <https://doi.org/10.1002/9780470612163.ch16Author Biography>

## **AUTHOR BIOGRAPHY**



**Xinyu Zhang** is a graduate student in computer technology at the University of Science and Technology Beijing. Her research mainly focuses on intelligence analysis based on big data and big data application. School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China



**Yang Li** is an Associate Professor at the Department of Computer Science and Technology, University of Science and Technology Beijing. Her research focuses on evolutionary computation and domain data engineering. School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China



**Genshen Chu** is a Ph.D. student at the University of Science and Technology Beijing. He has long been engaged in the research of massively parallel algorithms, high-performance scientific computing software and performance optimization. School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China