

Research and application of the parallel computing method for the grid-based Xin'anjiang model

Qian Liu^a, Dingsheng Wan^{a,*}, Yufeng Yu^a and Yangming Zhang^{a,b}

^a College of Computer and Information, Hohai University, Nanjing, Jiangsu 211100, China

^b Bank of Nanjing, Nanjing, Jiangsu 211100, China

*Corresponding author. E-mail: dshwan@hhu.edu.cn

ABSTRACT

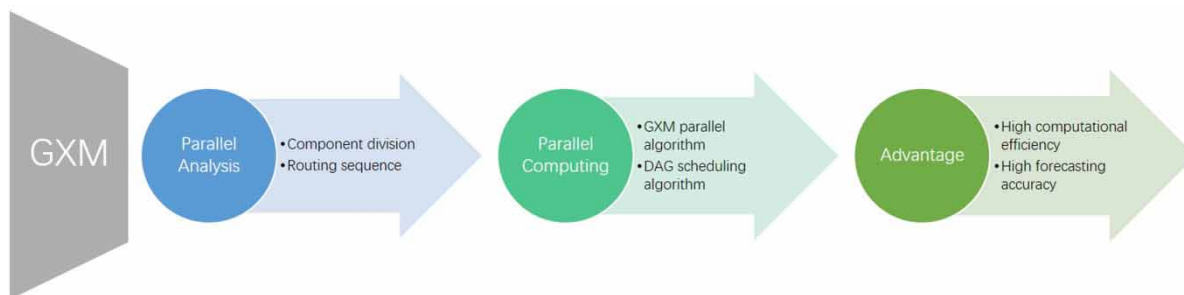
The grid-based Xin'anjiang model (GXM) has been widely applied to flood forecasting. However, when the model warm-up period is long and the amount of input data is large, the computational efficiency of the GXM is obviously low. Therefore, a GXM parallel algorithm based on grid flow direction division is proposed from the perspective of spatial parallelism, which realizes the parallel computing of the GXM by extracting the parallel routing sequence of the watershed grids. To solve data skew, a Directed Acyclic Graph (DAG) scheduling algorithm based on dynamic priority is proposed for task scheduling. The proposed GXM parallel algorithm is verified in the Qianhe River watershed of Shaanxi Province and the Tunxi watershed of Anhui Province. The results show that the GXM parallel algorithm based on grid flow direction division has good flood forecasting accuracy and higher computational efficiency than the traditional serial computing method. In addition, the DAG scheduling algorithm can effectively improve the parallel efficiency of the GXM.

Key words: distributed hydrologic model, flood forecasting, grid-based Xin'anjiang model, parallel computing

HIGHLIGHTS

- Proposing a parallel algorithm based on grid flow direction division, speeding up the computation of the GXM.
- Accurate flood forecasting facilitates flood management and flood control scheduling.
- DAG scheduling algorithm can effectively improve parallel efficiency.

GRAPHICAL ABSTRACT



INTRODUCTION

The hydrological model has always been a hot spot in hydrological research, which can be used for flood forecasting, hydrological mechanism research, hydrological and environmental process simulation and prediction. The hydrological model can provide decision support for flood control personnel, which is significant in flood control and disaster reduction. Hydrological models are divided into lumped and distributed hydrological models (Rui 2018). In the computation process of the lumped hydrological model, the spatial difference of the watershed is not considered and a set of unified and fixed parameters are adopted. Therefore, when the physical conditions of the underlying surface in the watershed vary greatly, the flood

This is an Open Access article distributed under the terms of the Creative Commons Attribution Licence (CC BY 4.0), which permits copying, adaptation and redistribution, provided the original work is properly cited (<http://creativecommons.org/licenses/by/4.0/>).

forecasting accuracy of the lumped hydrological model is not high enough. The distributed hydrological model takes into account the uneven distribution of rainfall in the watershed and the impact of the difference in the physical condition of the underlying surface (Rui 2017), so it has a high forecasting accuracy. Typical distributed hydrological models include the distributed Xin'anjiang model, a topography-based hydrological model (TOPMODEL) (Beven *et al.* 2021) and TOPKAPI model (Liu *et al.* 2005).

The distributed Xin'anjiang model can be divided into two types according to the partition granularity of the watershed: block-based distributed Xin'anjiang model and grid-based distributed Xin'anjiang hydrological model (GXM). The GXM is a distributed hydrological model based on the principle of the Xin'anjiang model and the digital elevation model (DEM) (Yao *et al.* 2009). GXM divides the watershed into high-precision grid units of the same size. The computation process of the GXM mainly includes evapotranspiration, runoff generation and water source division for each grid unit. Finally, according to the routing sequence of the upper and lower reaches of the watershed, the slope and channel confluence are computed grid by grid (Yao *et al.* 2021).

When GXM is computed serially, the computation time is long and computational efficiency is low, so GXM needs to be processed in parallel. The parallelization methods of distributed hydrological models are generally divided into parallelization in space, parallelization in time and parallelization in a combination of time and space (Ye *et al.* 2022). At present, the parallelization of distributed hydrological models mainly adopts the method of spatial parallelization. Liu *et al.* (2014) designed a parallel algorithm that stratifies the simulation units according to the flow direction for the distributed hydrological model Fully Sequential Dependent Hydrological Models (FSDHM). They parallelize the FSDHM by dividing simulation units (grids or sub-basins) without upstream and downstream dependencies into the same layer for simultaneous computation. Results have shown that the parallel algorithm can effectively improve the computational efficiency of FSDHM. Avesani *et al.* (2021), based on the Message Passing Interface (MPI) standard, designed a parallel hydrological model, named HYPERstreamHS. The model adopts a dual-layer parallelization strategy to fully exploit both spatial domain and parameters domain decomposition, and reduce the computational time by utilizing the available processors efficiently.

In this study, we proposed a GXM parallel algorithm based on grid flow direction division, which realizes the parallel computation of the GXM confluence process and improves the computation speed of the model. Aiming at the possible data skew in the computation process, a DAG scheduling algorithm based on dynamic priority was designed, which realizes task scheduling and improves the parallel efficiency in the computation process.

The following content of this article is a description of the GXM and parallel algorithm; data and research used in this experiment; a description and discussion of experimental results; finally, a summary of this research.

METHODS

Grid-based Xin'anjiang model

The GXM uses high-precision grids as the basic simulation unit. First, the three-layer evapotranspiration component and the runoff generation component are used to calculate the evapotranspiration and runoff of each grid unit. Then, the free water reservoir structure is used to obtain surface runoff, subsurface runoff and underground runoff from each grid unit. Finally, according to the Muskingum method, the slope and river confluence are computed grid by grid. The surface runoff, subsurface runoff and underground runoff of the grid in the watershed are computed to the outlet of the watershed (Li *et al.* 2007).

According to functional differences, the computing modules in the GXM are divided into four components: evapotranspiration component, runoff generation component, water source division component and flow routing component. The component divisions are shown in Figure 1. According to the characteristics of the component, components are divided into computation-independent and space-dependent components (Liu *et al.* 2013a). The grid simulation units are independent of each other in the computation process of computation-independent components. In other words, the computation-independent components have no dependencies between the upper and lower reaches of the watershed. The independent components include evapotranspiration, runoff generation and water source division components. In the computation process of space-dependent components, including flow routing components, the grid simulation units need to consider the upstream and downstream dependencies.

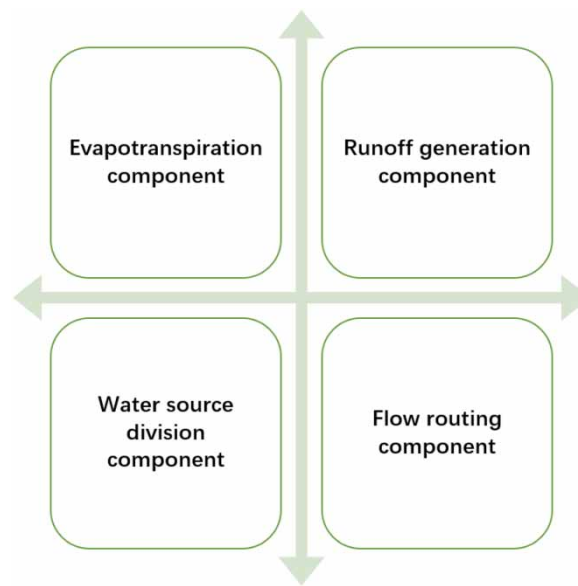


Figure 1 | Component division of the GXM.

GXM parallel algorithm based on grid flow direction division

The most essential condition for components to realize parallel computing is that some grid units are independent and do not affect each other during the computation process (Liu *et al.* 2016). Therefore, the computation of independent components, including evapotranspiration, runoff generation and water source division components, is naturally parallelizable.

Computation-independent components only have time dependencies. In other words, in the cyclic computation process of a single period, the grid units in the watershed are independent, and there is no data exchange between grids. Therefore, we can evenly divide the simulation grid units into several parts for parallel calculation. However, when we parallelize space-dependent components, we cannot directly process grid units in parallel like computation-independent components. Because there is not only time dependence between grid simulation units, but also space dependence in the computation process of space-dependent components.

For example, when we use the Muskingum-based grid-by-grid calculation method to compute the channel runoff according to the grid flow direction and finally obtain the outflow of the outlet grid in the watershed. As shown in Figure 2, when the flow Q_d is computed, the upstream inflow Q_a , Q_b and Q_c all need to be considered.

Therefore, we proposed a GXM parallel algorithm based on grid flow direction. By constructing the grid flow direction matrix, cumulative catchment area matrix and the water system matrix of the watershed, the flow relationship between the grids in the watershed can be decomposed to find out the parallel routing sequence of the grid simulation units in the watershed. This way, grid unit sets with the same routing sequence are computed in parallel to realize the parallelization of flow routing components.

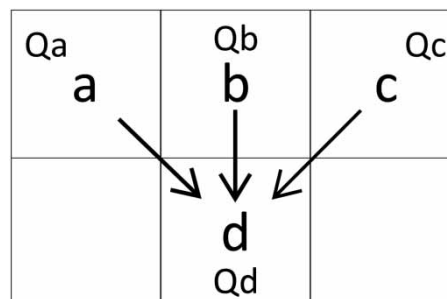


Figure 2 | Schematic diagram of grid-by-grid confluence based on the Muskingum method.

The D8 algorithm (Wu *et al.* 2006) is used to determine the water flow direction of the grid unit. The idea of the D8 algorithm is as follows: there are only eight possibilities for the water flow direction of each grid unit and every grid's flow only flows into one of the eight adjacent grids. Different numbers represent the eight flow directions of the grid units. The flow direction marks of the grid units are shown in Figure 3.

In the 3 × 3 grid window, first, the distance weight difference between the central grid and each adjacent grid is calculated. The adjacent grid with the largest distance weight difference is set as the outflow grid of the central grid. The formula for distance weight difference between grids is shown as:

$$\tan(\theta) = \frac{Z_i - Z_j}{\alpha * D} \tag{1}$$

Among them, Z_i represents the grid elevation whose flow direction needs to be determined. Z_j represents the elevation of the adjacent grid. α refers to the distance weight. The α of the diagonal grids is $\sqrt{2}$ and the α of other positive grids is 1. D is the side length of the grid. Figure 4 shows a schematic diagram of the grid flow direction matrix and the corresponding water flows direction matrix.

The grid flow direction matrix can determine the upstream grid of all grid cells in the watershed. The runoff in the watershed can be computed grid by grid to the outlet of the watershed according to the flow direction of the grid flow direction matrix. The cumulative catchment area matrix stores the cumulative catchment of each grid unit in the watershed and

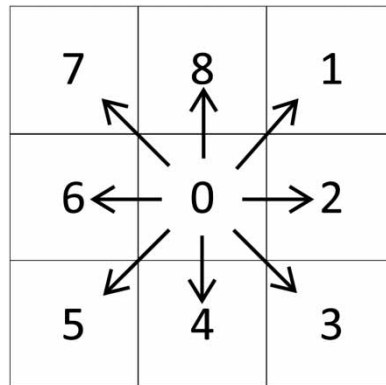


Figure 3 | Grid flow direction markers.

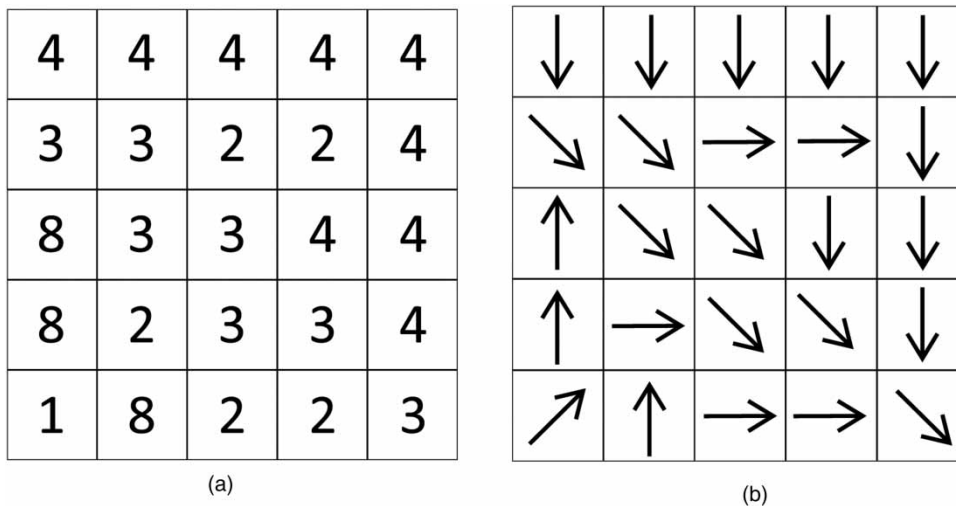


Figure 4 | Grid flow matrix (a) and water flow matrix (b).

each value in the matrix represents the number of grid units flowing into the current grid from the upper reaches. When the catchment area of the grid exceeds a certain threshold, the grid unit is set as a channel grid and marked as 1. When the catchment area of the grid is below the threshold, the grid is set as a non-channel grid and marked as 0. Finally, these marked grid cells form the water system matrix. The process of generating a cumulative catchment area matrix and water system matrix through a grid flow direction matrix is shown in Figure 5.

To decouple the computational relationship between the grids, we proposed the multi-level parallel grid division algorithm, which can obtain the parallel routing sequence of the grids in the watershed to realize the parallel computation of the GXM. The algorithm is described as follows:

- (1) First, the routing order of the outlet grid in the watershed is initialized. We label the number of grids in the watershed as N and the catchment area of each grid as S . The grid (outlet grid of watershed) coordinate (i, j) with the largest catchment area (S_{max}) is searched according to the catchment area matrix. We set the routing order of this grid to $n(n = 1)$.
- (2) All upstream grid coordinates of each grid in the watershed are computed. The eight adjacent points around the current grid point are scanned clockwise and represented by a set T :

$$T = \{(i - 1, j - 1), (i - 1, j), \dots, (i + 1, j + 1)\} \tag{2}$$

- (3) To compute the routing order of upstream grids, we find out the upstream grids of the current grid according to the grid flow direction. If the adjacent grid flows into the current grid, the routing order of the adjacent grid is set to $n + 1$.
- (4) After the routing order of all adjacent grids are computed, the adjacent grid is updated as the current grid. We repeat the above steps continuously until the adjacent grids cannot be found.
- (5) We reverse the routing sequence obtained and then traverse all grids in the watershed. If the grid belongs to a water system, the routing order of the grid is retained. If the grid does not belong to a water system, the routing order of the grid is initialized to 1. Finally, the routing order of grids belonging to the water system is obtained.
- (6) Initialize the routing order of slope grids: if the current grid is a slope grid, we judge whether the current grid has upstream and downstream grids according to the catchment area. If the grid has no upstream and downstream grids, the routing order of the grid is initialized to $n(n = 1)$.
- (7) To compute the routing order of slope grids, we find the downstream slope grids according to the grid flow matrix and add 1 to the maximum routing order of all upstream grids.
- (8) After the routing order of the adjacent grid is computed, the adjacent grid is updated to the current grid. Then we repeat the above steps until the routing order of all slope grids in the watershed is obtained.

Through the above algorithm, we can obtain the parallel grid routing sequence in the watershed. Assuming that the number of grids in the watershed is N , the maximum routing order is Max , the routing order is i and the number of repeated grids with routing order i is C_i , the relationship between them is shown as follows:

$$N = \sum_{i=1}^{Max} C_i \tag{3}$$

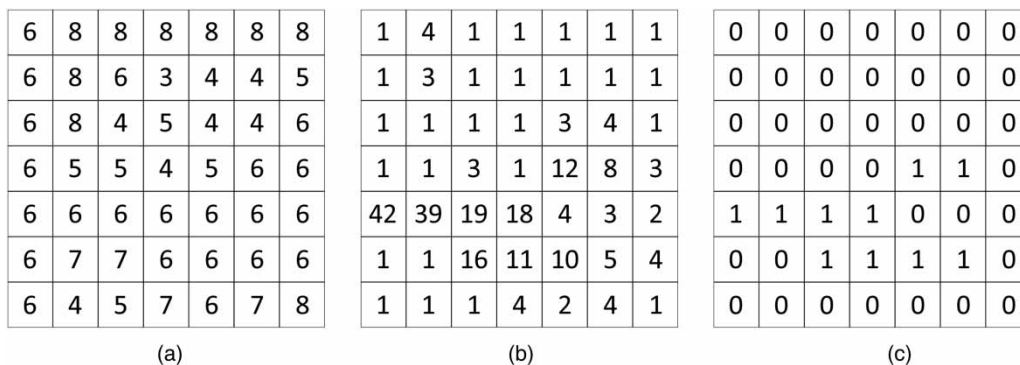


Figure 5 | Process of generation of cumulative catchment area matrix (b) and water system matrix (c) by grid flow matrix (a).

These grid units with the same routing order can be computed in parallel because these grid units do not have upstream and downstream dependencies. The grid units of each routing order are counted to obtain the distribution table of the parallel routing order of the watershed. Table 1 shows the distribution table of the parallel routing order of the Tunxi watershed in Anhui Province. We can find that as the routing order increases, the number of slope grids decreases gradually, and the routing order of the watershed outlet grid is the largest.

DAG scheduling algorithm based on dynamic priority

Data skew is prone to occur between threads when the amount of input data is large. Therefore, we propose a DAG scheduling algorithm based on dynamic priority. We generate a DAG to represent computational relationships of grids in the watershed by connecting valid grids in the watershed according to the parallel routing order. First of all, the channel grids are used as the initial critical path. Second, we establish three task status queues including the ready queue, the running queue, the completed queue and put task nodes to be executed into the ready queue, and running task nodes into the running queue.

We divide task nodes in the ready queue into three types: leaf node, waiting node, and critical node. The leaf nodes are at the first layer in the DAG graph whose task priority is initialized to 1. Waiting nodes are nodes that will be executed. Critical nodes on the critical path have the largest sum of layers and depths and the highest priority. The task priority of the computing nodes and critical path are dynamically updated in the computational process. So, the remaining tasks can be scheduled continually according to the updated task priority. The schematic diagram of task scheduling in the process of GXM parallel computing is shown in Figure 6.

Table 1 | Distribution table of watershed parallel routing order in the Tunxi Basin

Calculation order	Channel grid	Slope grid	Amount of grid
1	0	1872	1,871
2	0	493	615
...
99	1	0	1
100	1	0	1

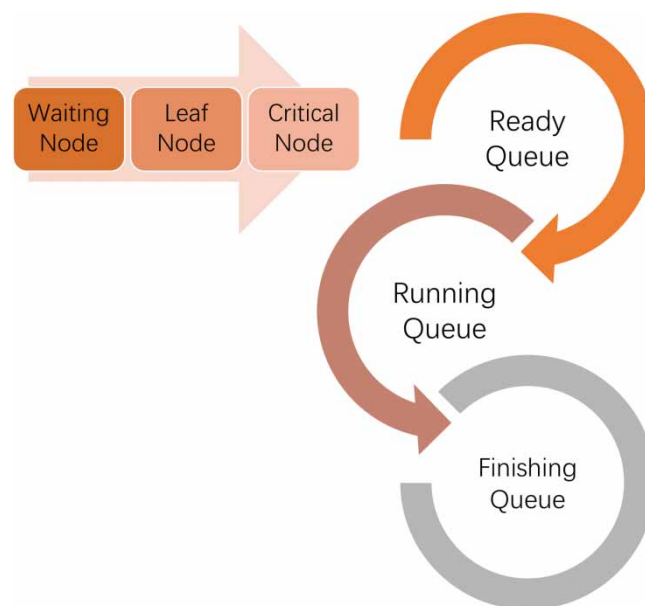


Figure 6 | Task scheduling in the GXM parallel computing process.

Model performance measures

Liu *et al.* (2013b) used the theoretical maximum speedup ratio (TMSR) to measure the performance of model parallelization, that is, the ratio of the computation time of the serial algorithm to the computation time of the parallel algorithm, which is used as an evaluation index of the parallel computing ability of the distributed hydrological model. We use the ratio of the speedup ratio to the number of threads to measure the parallel efficiency of the scheduling algorithm. We use the Nash–Sutcliffe coefficient (NSE), root-mean-square error (RMSE) (Kang *et al.* 2022) and relative error (RE) to measure the flood simulation accuracy of the GXM. The formulas of NSE, RMSE and RE are as follows:

$$\text{NSE} = 1 - \frac{\sum_{i=1}^n (y_i - \tilde{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (4)$$

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \tilde{y}_i)^2} \quad (5)$$

$$\text{RE} = \frac{\sum_{i=1}^n (\tilde{y}_i - y_i)}{\sum_{i=1}^n y_i} \times 100\% \quad (6)$$

where y_i , \tilde{y}_i , \bar{y} represent the observed value, simulated value and mean value, respectively. NSE ranges from 1 to $-\infty$. When NSE is close to 1, it means that the prediction result of the model has higher reliability. Conversely, when the gap between NSE and 1 is larger, it indicates that the prediction result of the GXM is not credible. RMSE ranges from 0 to $+\infty$ and can reflect the accuracy of the prediction of the model. When RMSE is close to 0, the prediction results of the model are more accurate. RE ranges from -100% to $+100\%$. The optimum value of RE is 0, which indicates that the prediction results are accurate.

DATA AND STUDY DOMAIN

Tunxi is located in the south of Anhui Province, with a basin area of 2,696 km². It is a mountainous river located in a subtropical monsoon climate zone. The watershed is rich in vegetation, with a forest coverage rate of over 70%. Precipitation in the Tunxi Basin is mainly concentrated in the flood season, which accounts for more than 60% of the total annual precipitation. The main types of rainstorms are frontal and low-pressure rainstorms. One medium-sized reservoir in the basin has a storage capacity of 29 million m³. In the basin's upper reaches, 43 small reservoirs have been built, with a total storage capacity of 22 million m³. Considering the distribution of hydrological stations, 18 hydrological stations and one evaporation station were selected for flood simulation in the Tunxi basin. The water system distribution in the Tunxi Basin is shown in Figure 7.

Qianhe River, located in the south of Shanxi Province, belongs to the semi-arid basin, and the basin area of Qianhe River is 2,935 km². The terrain is higher in the northwest and lowers in the southeast, the annual average rainfall is 696 mm. The yearly rainfall distribution in the Qianhe River Basin is uneven, mainly concentrated from June to September. The regional distribution of precipitation in the basin is primarily affected by climate, geographical and environmental factors and the precipitation gradually increases from north to south.

In the Qianhe River Basin, 13 hydrological stations were selected for flood simulation. The distribution of the water system in the Qianhe River Basin is shown in Figure 8.

The vegetation leaf index, vegetation height and other parameters required by the GXM can be directly assigned according to the vegetation type of the grid unit by using the existing research conclusions. Parameters such as tension water storage capacity, free water storage capacity, soil inflow and subsurface runoff outflow coefficients can be estimated through their quantitative relationships with features such as topographic index, soil type, and soil layer thickness. The area proportion method based on the flow direction can estimate the proportion parameter of surface runoff entering the channel. On this

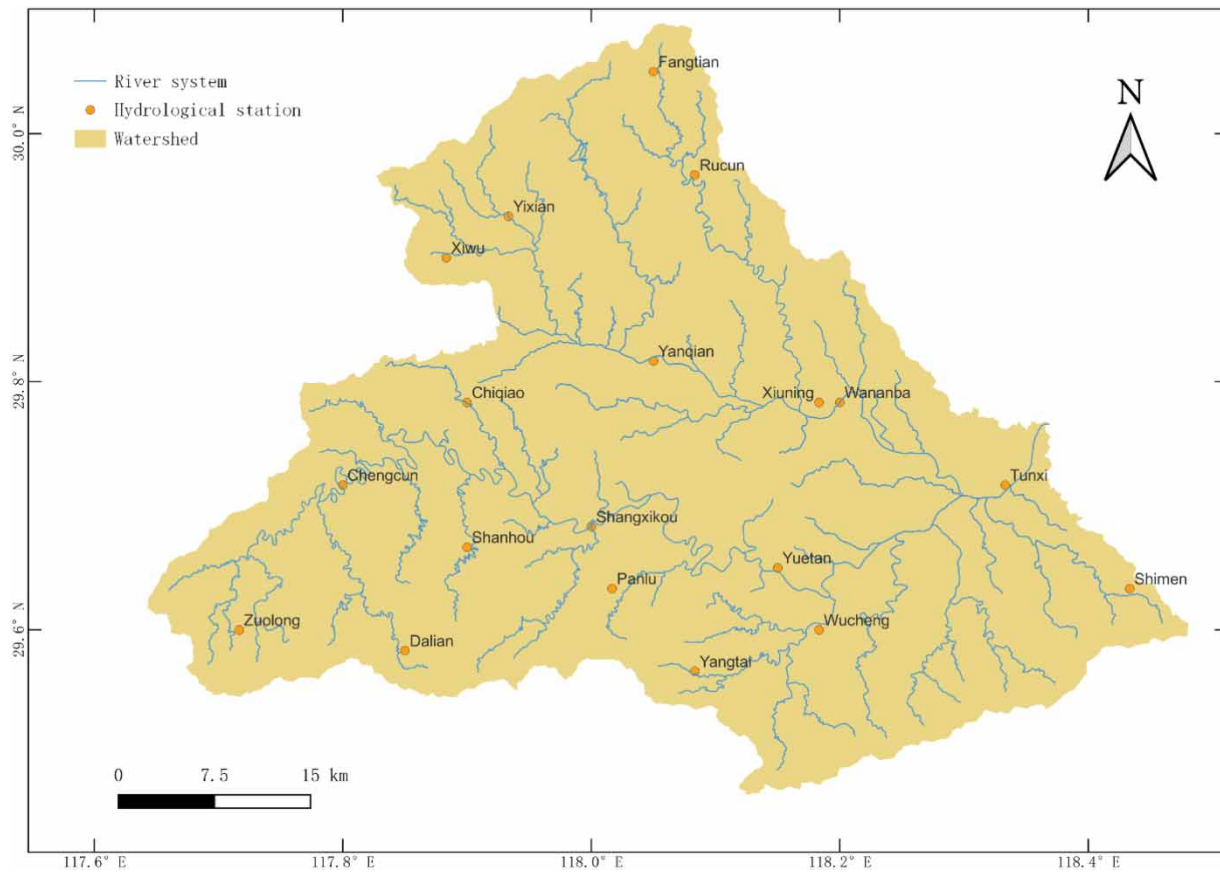


Figure 7 | Location of hydrological stations in the Tunxi Basin.

basis, the observed hydrological data are used to calibrate and estimate the remaining parameters such as the full storage rate, and finally, the model parameter set of each grid unit can be obtained (Yao *et al.* 2021).

The underlying surfaces of the Tunxi Basin and the Qianhe River Basin use grid data with a resolution of $1 \text{ km} \times 1 \text{ km}$. We selected the rainfall and evaporation data of the Tunxi Basin in July 2020 and June 2016 and selected the rainfall and evaporation data of the Qianhe River Basin in July 2010, July 2018 and July 2016. We directly use the real-time data of surface rainfall stations, and use Thiessen polygon, inverse distance squared and other methods for spatial interpolation to obtain rainfall data in the watershed. Watershed evaporation data are computed from evaporation station data in the watershed.

RESULTS AND DISCUSSION

Efficiency analysis of parallel algorithms

In order to verify the computational efficiency of the GXM parallel algorithm, we first set the warm-up period from 3 to 30 days in July 2018 in the Qianhe River Basin and compared the computation time of the serial algorithm and the GXM parallel algorithm based on grid flow direction division. It can be seen from Figure 9 that the speedup ratio of GXM was 1.02 when the warm-up period is 3 days and the speedup ratio was 2.78 when the warm-up period is 15 days. With the increase of the warm-up period, the parallel speedup ratio of the GXM was constantly improving and the computation time of the serial algorithm increases significantly. When the warm-up period increases to 30 days, the parallel speedup reached 4.01, and the computation time of the serial algorithm reached 800 s approximately but the computation time of the parallel algorithm was only 198 s.

Then, we continued to conduct a computation time comparison experiment in the Tunxi Basin. We set the warm-up period from 3 to 30 days in July 2020 and compared the model computation time of the serial algorithm and the GXM parallel algorithm. It can be seen from Figure 10 that when the warm-up period reaches 3 days, the GXM speedup ratio was 1.03 and when

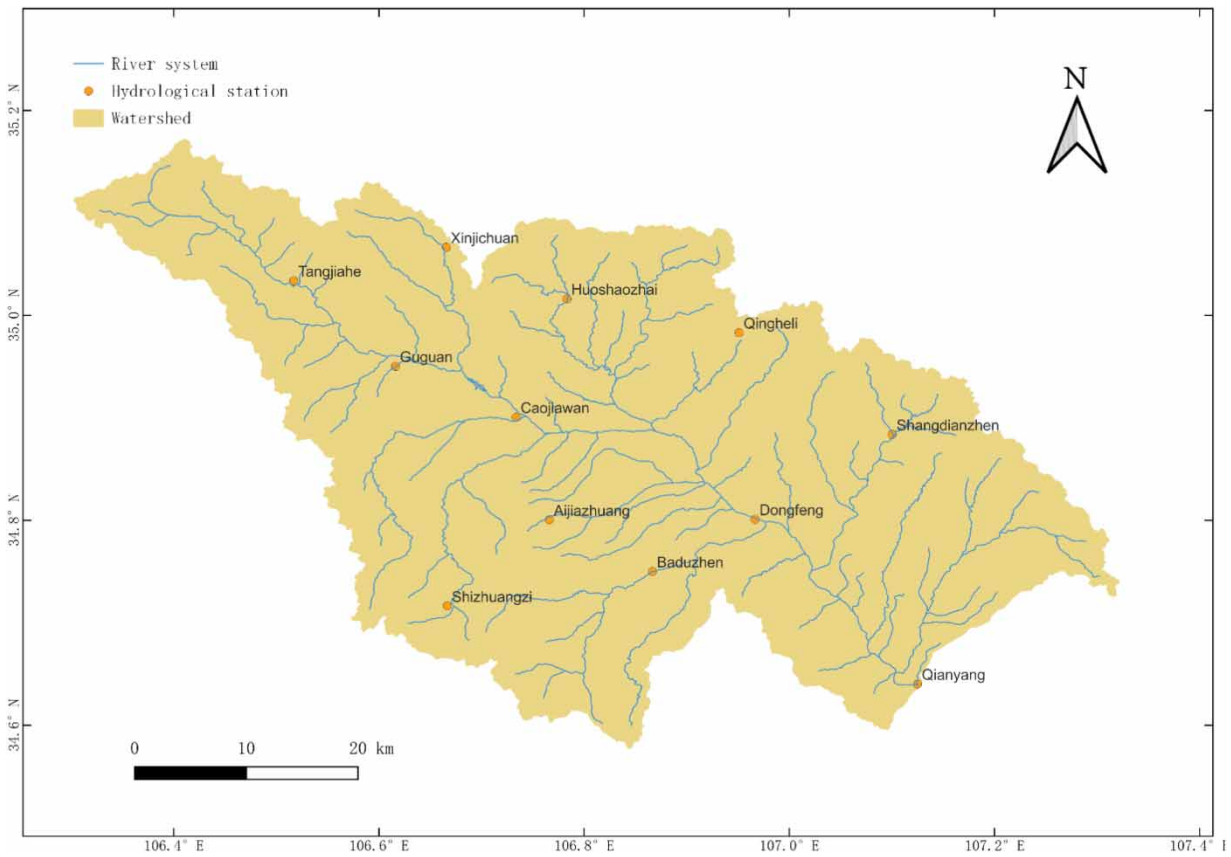


Figure 8 | Location of hydrological stations in the Qianhe River Basin.

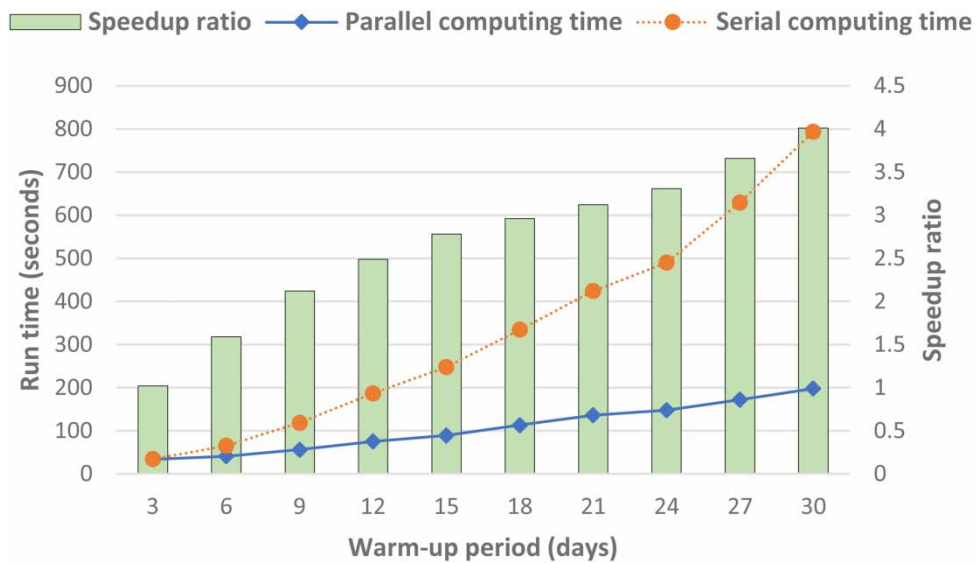


Figure 9 | Computation time comparison between the serial and parallel algorithm in the Qianhe River.

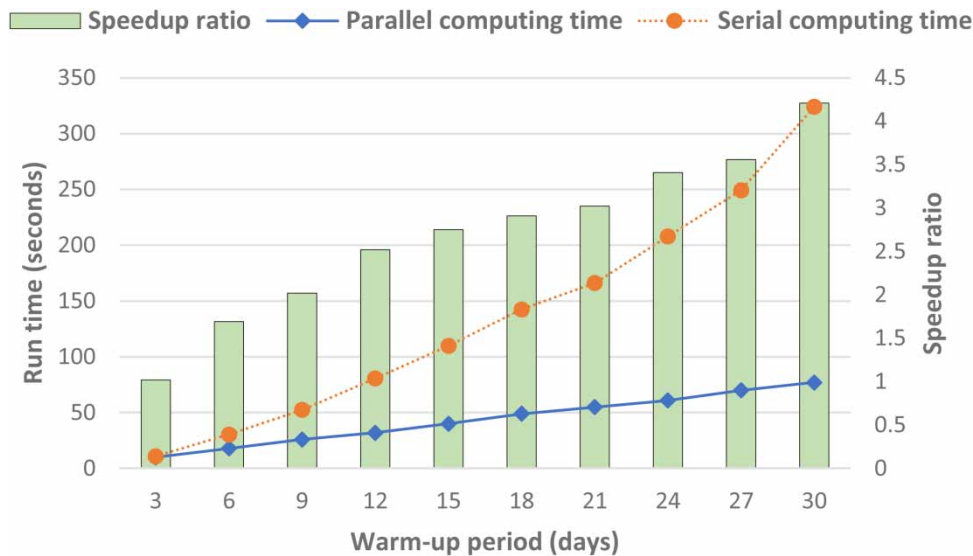


Figure 10 | Computation time comparison between the serial and parallel algorithm in the Tunxi Basin.

the warm-up period reaches 15 days, the speedup ratio reached 2.75. When the warm-up period increases to 30 days, the GXM parallel speedup reached 4.21.

From the above two comparison experiments of computation time, we can draw the following conclusions: compared with the original serial algorithm, the computational efficiency of the GXM parallel algorithm based on grid flow direction division has been significantly improved. As the amount of model input data continues to increase, the acceleration performance of parallel algorithms becomes more and more significant.

Qin *et al.* (2020) realized the parallelization of the confluence process of the WEP-L distributed hydrological model by designing a domain decomposition algorithm and implemented the task scheduling based on the greedy algorithm. The algorithm decomposes the whole watershed into the process of mainstream confluence and tributary stream confluence. Since the tributary hydrological areas are independent of each other and there is no confluence relationship between the tributary areas, each tributary hydrological area can be assigned to different threads to achieve parallel computing. It is verified by experiments that when the number of computing threads is 4, the speedup ratios of the tributary confluence process and the main branch confluence process both reach the maximum, which are 3.04 and 2.51, respectively. Compared with the above algorithm, the GXM parallel algorithm proposed in this paper has a finer granularity for the division of the underlying surface, and parallelizes the GXM by extracting the parallel routing sequence of the underlying surface grids of the watershed, thus having higher computational efficiency.

We set the warm-up period of GXM to 60 days in the Tunxi Basin and adopted different numbers of threads to verify the parallel efficiency (Zhang *et al.* 2016) of the DAG scheduling algorithm based on dynamic priority. Figure 11 shows that as the number of threads increased, parallel efficiency slowly degraded because the overhead for inter-thread communication continues to increase. However, the parallel efficiency after using the task scheduling algorithm was always higher than the original parallel algorithm. The result shows that the scheduling algorithm can improve the resource utilization of the computer and reduce the unbalanced thread load.

Accuracy analysis of the GXM parallel algorithm in flood forecasting

We selected a total of four flood events in the Tunxi Basin and the Qianhe River Basin for discharge simulation to verify the flood forecasting accuracy of the GXM parallel algorithm.

The four flood events are:

- (1) Flood event 1 (Figure 12) that occurred in the Tunxi Basin from 21:00 on 1 July 2020 to 22:00 on 13 July 2020.
- (2) Flood event 2 (Figure 13) that occurred in the Tunxi Basin from 21:00 on 17 June 2016 to 21:00 on 24 June 2016.

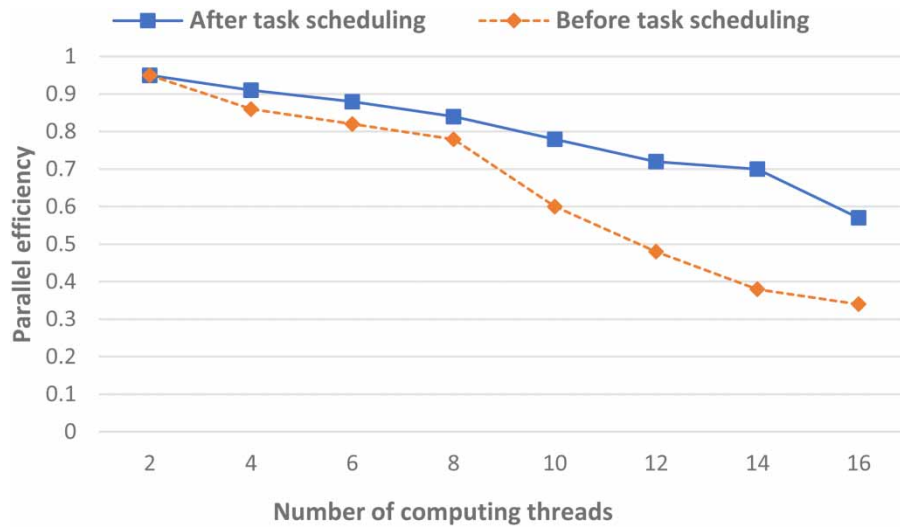


Figure 11 | Parallel efficiency comparison.

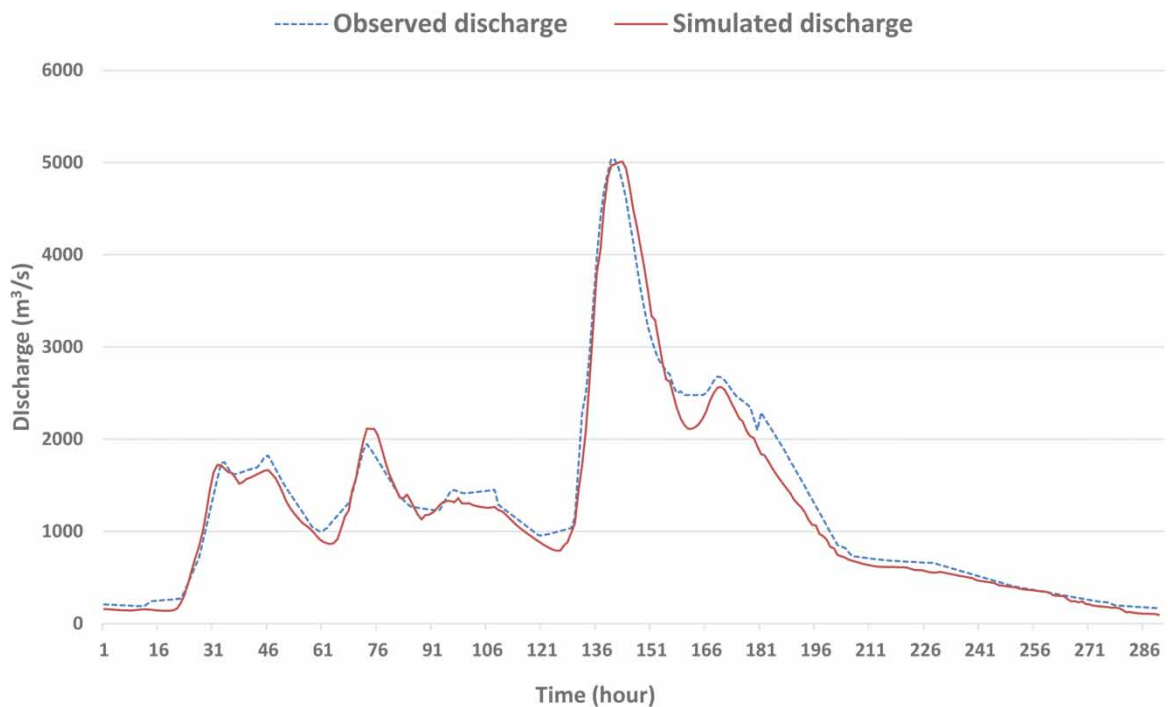


Figure 12 | The simulated hydrographs in the Tunxi Basin during 1 July–13 July 2020.

- (3) Flood event 3 (Figure 14) that occurred in the Qianhe River Basin from 05:00 on 23 July 2010 to 18:00 on 27 July 2010.
 (4) Flood event 4 (Figure 15) that occurred in the Qianhe River Basin from 00:00 on 10 July 2018 to 14:00 on 13 July 2018.

The model warm-up period of the four flood simulation experiments is set to 30 days. The resolution of the underlying surface of the two watersheds is $1 \text{ km} \times 1 \text{ km}$. There are 3,619 effective grids in the Tunxi Basin and 11,448 effective grids in the Qianhe River Basin. The evaluation indicators of model simulation performance are NSE, RMSE and RE.

Figure 12 shows the observed and simulated hydrograph of flood event 1. The peak time of the observed discharge is at 16:00 on 7 July 2020. At this time, the observed discharge is $5,040 \text{ m}^3/\text{s}$, the simulated discharge calculated by the GXM is $4,968 \text{ m}^3/\text{s}$ and the error value of the simulated flow is $72 \text{ m}^3/\text{s}$. The flood peak time calculated by the GXM is 19:00

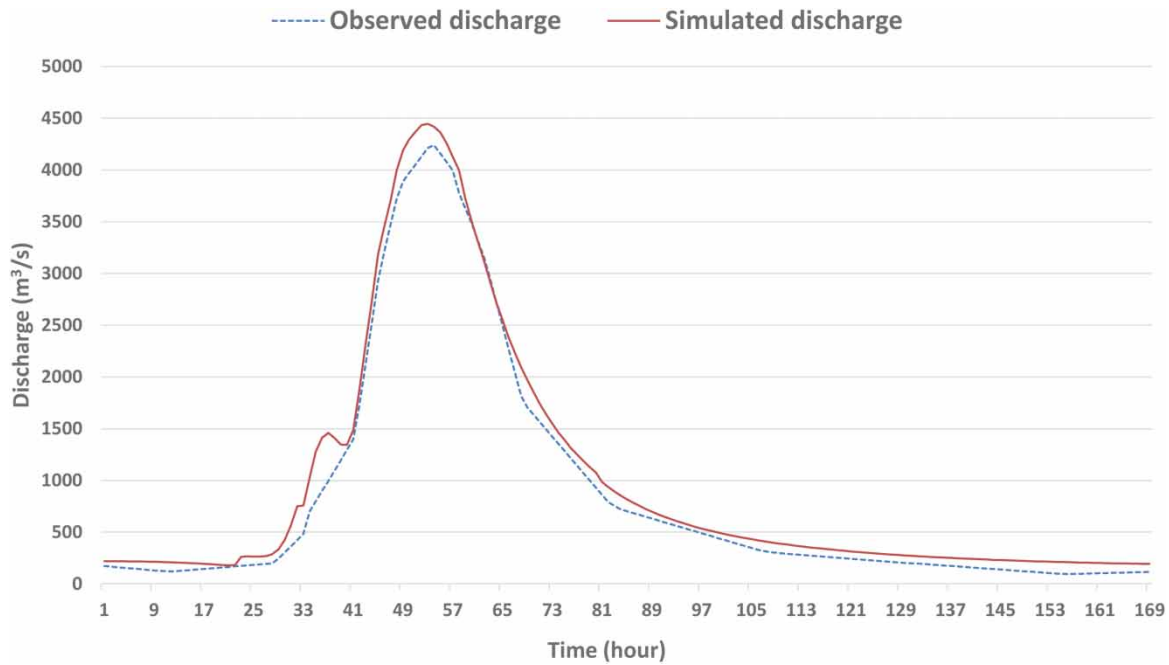


Figure 13 | The simulated hydrographs in the Tunxi Basin during 17 June–24 June 2016.

on 7 July 2020. At this time, the observed discharge is 4,790 m³/s, the simulated discharge calculated by the GXM is 5,010 m³/s and the error value of the simulated discharge is 220 m³/s. At this time, the NSE of the GXM is 0.9745 and the RMSE of the GXM is 164.89 m³/s.

Figure 13 shows the observed and simulated hydrograph of flood event 2. The peak time of the observed discharge is at 02:00 on 20 June 2016. At this time, the observed discharge is 4,239 m³/s, the simulated discharge calculated by the

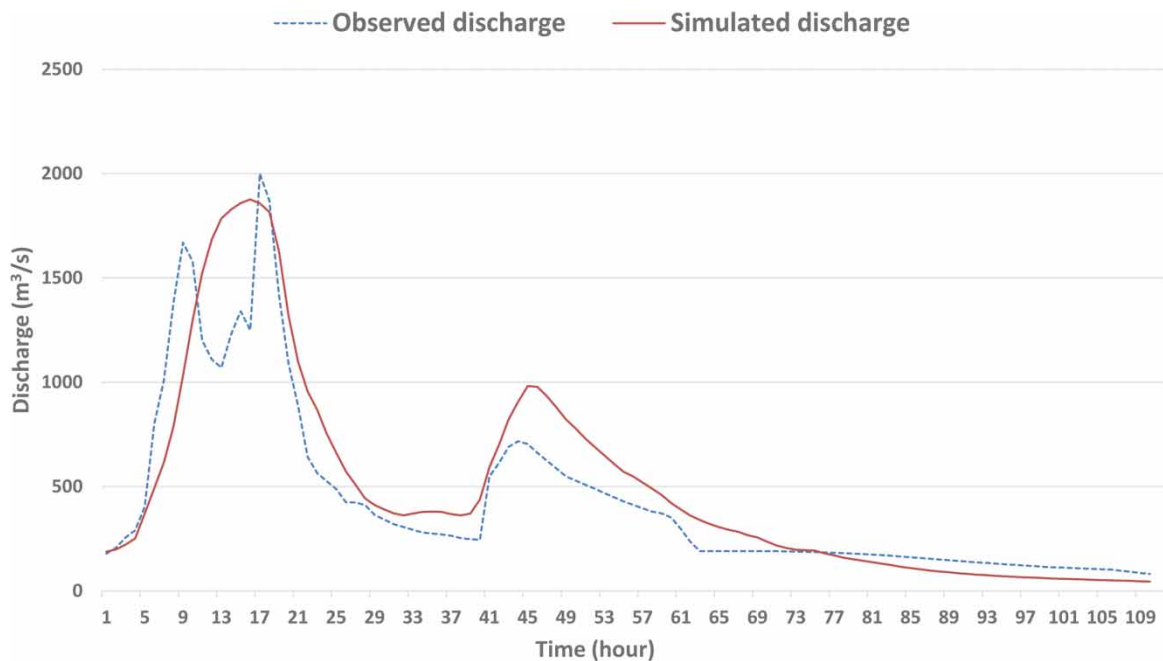


Figure 14 | The simulated hydrographs in the Qianhe River Basin during 23 July–27 July 2010.

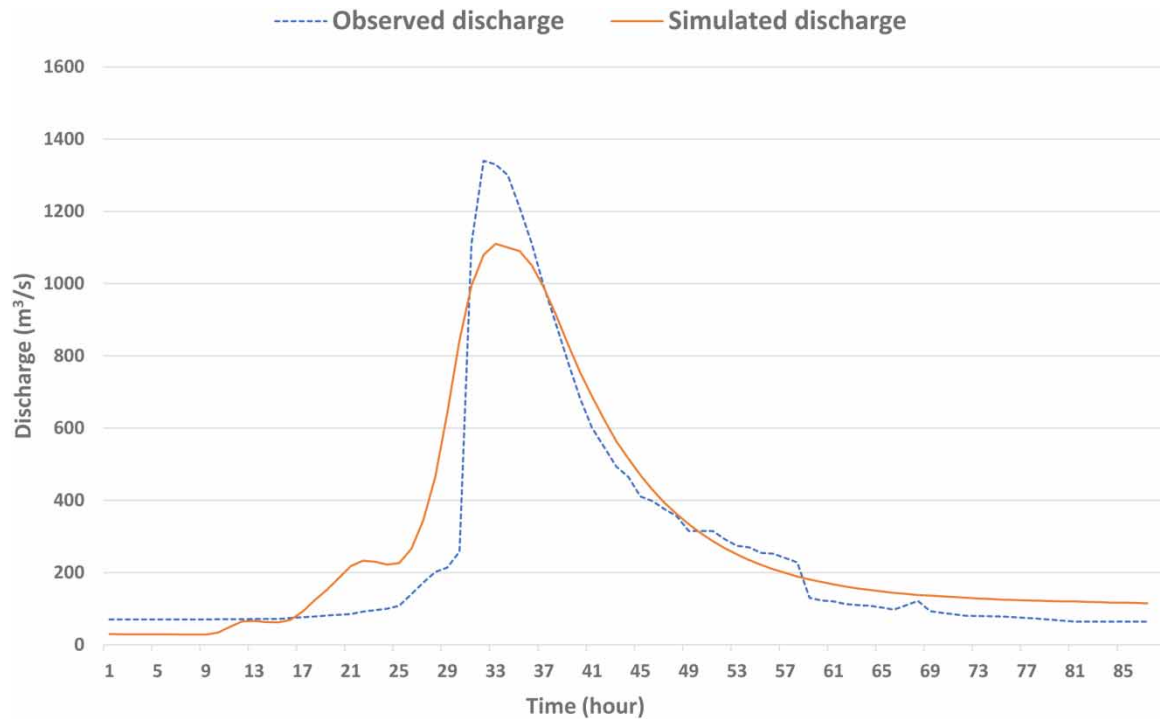


Figure 15 | The simulated hydrographs in the Qianhe River Basin during 10 July–13 July 2018.

model is $4,417 \text{ m}^3/\text{s}$ and the error value of the simulated flow is $178 \text{ m}^3/\text{s}$. The flood peak time calculated by the model is 01:00 on 20 June 2016. At this time, the observed discharge is $4,213 \text{ m}^3/\text{s}$, the simulated discharge calculated by the model is $4,446 \text{ m}^3/\text{s}$ and the error value of the simulated discharge is $233 \text{ m}^3/\text{s}$. At this time, the NSE of the GXM is 0.9850 and the RMSE of the GXM is $141.28 \text{ m}^3/\text{s}$.

Figure 14 shows the observed and simulated hydrograph of flood event 3. The peak time of the observed discharge is at 21:00 on 23 July 2010. At this time, the observed discharge is $2,000 \text{ m}^3/\text{s}$, the simulated discharge calculated by the model is $1,857 \text{ m}^3/\text{s}$ and the error value of the simulated flow is $143 \text{ m}^3/\text{s}$. The flood peak time calculated by the model is 20:00 on 23 July 2010. At this time, the observed discharge is $1,250 \text{ m}^3/\text{s}$, the simulated discharge calculated by the model is $1,876 \text{ m}^3/\text{s}$ and the error value of the simulated discharge is $626 \text{ m}^3/\text{s}$. At this time, the NSE of the GXM is 0.8230 and the RMSE of the GXM is $205.06 \text{ m}^3/\text{s}$.

Figure 15 shows the observed and simulated hydrograph of flood event 4. The peak time of the observed discharge is at 07:00 on 11 July 2018. At this time, the observed discharge is $1,340 \text{ m}^3/\text{s}$, the simulated discharge calculated by the model is $1,080 \text{ m}^3/\text{s}$ and the error value of the simulated flow is $260 \text{ m}^3/\text{s}$. The flood peak time calculated by the model is 08:00 on 11 July 2018. At this time, the observed discharge is $1,330 \text{ m}^3/\text{s}$, the simulated discharge calculated by the model is $1,110 \text{ m}^3/\text{s}$ and the error value of the simulated discharge is $220 \text{ m}^3/\text{s}$. At this time, the NSE of the GXM is 0.8853 and the RMSE of the GXM is $110.34 \text{ m}^3/\text{s}$.

The evaluation indexes of simulated results are shown in Table 2. It can be found that the GXM has achieved good flood simulation accuracy in the Tunxi and Qianhe River Basins, which makes it greatly practical in flood management and utilization. However, the simulation accuracy of the GXM in the Qianhe River Basin was lower than that in the Tunxi River Basin. By comparing the hydrographs of the above four flood events, it can be clearly found that the shape of the hydrographs in the Qianhe River Basin fluctuated more obviously and the difference in flood peak discharge was larger. The reason for the flood simulation results is mainly because the GXM is mainly suitable for humid and semi-humid areas, while the Qianhe River Basin is a semi-arid area. The formation process of rainfall runoff in semi-arid regions is more complicated than that in humid regions, with both full-storage runoff and excess infiltration runoff, which brings great difficulties to runoff simulation.

Table 2 | Performance of the GXM in the Tunxi Basin and the Qianhe River Basin

Flood event	RE (%)	Peak time difference (h)	NSE	RMSE (m/s)
Event 1 in Tunxi	-5.909	3	0.9754	164.89
Event 2 in Tunxi	13.167	-1	0.9850	141.27
Event 3 in Qianhe River	14.518	-1	0.823	205.06
Event 4 in Qianhe River	11.861	1	0.8853	110.34

CONCLUSIONS

This study proposes a GXM parallel algorithm based on grid flow direction division and a DAG scheduling algorithm based on dynamic priority to realize the parallel computing and task scheduling of the GXM. The GXM parallel algorithm and the DAG scheduling algorithm were applied in the Tunxi Basin and the Qianhe River Basin, the results have shown that the GXM parallel algorithm can significantly improve the computational efficiency of the GXM and the DAG scheduling algorithm can increase parallel efficiency and reduce resource overhead obviously. In addition, four flood simulation experiments were conducted in the Tunxi Basin and the Qianhe River Basin, the simulation results show that the GXM parallel algorithm has good flood forecasting accuracy and is useful for flood management.

In recent years, the graphics processing unit (GPU) has been widely used in the parallel computing of hydrological models. Freitas *et al.* (2022) used a CPU + GPU system to realize the parallelization of the Modelo de Grandes Bacias (MGB) hydrological model and achieved a significant performance improvement. In the future, we will continue to study how to reduce the communication overhead of the GXM between grids during parallel computing to further improve the computational efficiency of the GXM, and try to realize the parallelization of the GXM by utilizing GPU.

ACKNOWLEDGEMENT

This study was supported by the National Key Research and Development Program of China (2021YFB3900605) and the National Key Research and Development Program of China (2018YFC1508106).

DATA AVAILABILITY STATEMENT

Data cannot be made publicly available; readers should contact the corresponding author for details.

CONFLICT OF INTEREST

The authors declare there is no conflict.

REFERENCES

- Avesani, D., Galletti, A., Piccolroaz, S., Bellin, A. & Majone, B. 2021 A dual-layer MPI continuous large-scale hydrological model including Human Systems. *Environmental Modelling & Software* **139**, 105003.
- Beven, K. J., Kirkby, M. J., Freer, J. E. & Lamb, R. 2021 A history of TOPMODEL. *Hydrology and Earth System Sciences* **25** (2), 527–549.
- Freitas, H. R., Mendes, C. L. & Ilic, A. 2022 Performance optimization of the MGB hydrological model for multi-core and GPU architectures. *Environmental Modelling & Software* **148**, 105271.
- Kang, D., Lee, K. & Kim, E. 2022 Improvement of springtime streamflow prediction using a snow hydrology model aided with USDA SNOTEL and in-situ snowpit observations. *Hydrology Research* **53** (12), 1510–1528.
- Li, Z. J., Yao, C. & Wang, Z. H. 2007 Development and application of grid-based Xinanjiang model. *Journal of Hohai University* **35** (2), 131–134.
- Liu, Z., Martina, M. L. & Todini, E. 2005 Flood forecasting using a fully distributed model: application of the TOPKAPI model to the Upper Xixian Catchment. *Hydrology and Earth System Sciences* **9** (4), 347–364.
- Liu, J. Z., Zhu, A. X., Qin, C. Z., Chen, L. J., Wu, H. & Jiang, J. C. 2013a Review on parallel computing of distributed hydrological models. *Progress in Geography* **32**, 538–547.
- Liu, J., Zhu, A. X. & Qin, C. Z. 2013b Estimation of theoretical maximum speedup ratio for parallel computing of grid-based distributed hydrological models. *Computers & Geosciences* **60**, 58–62.
- Liu, J., Zhu, A. X., Liu, Y., Zhu, T. & Qin, C. Z. 2014 A layered approach to parallel computing for spatially distributed hydrological modeling. *Environmental Modelling & Software* **51**, 221–227.

- Liu, J., Zhu, A. X., Qin, C. Z., Wu, H. & Jiang, J. 2016 A two-level parallelization method for distributed hydrological models. *Environmental Modelling & Software* **80**, 175–184.
- Qin, Z. N., Li, S., Zhou, Z. H., Liu, M. T., Mou, D., Liu, J. J. & Xiang, D. 2020 Domain decomposition parallel computing method of distributed hydrological model and its application. *Water Resources and Power* **38** (10), 1–4 + 12.
- Rui, X. F. 2017 Discussion of watershed hydrological model. *Advances in Science and Technology of Water Resources* **37** (04), 1–7 + 58.
- Rui, X. F. 2018 More discussion of watershed hydrological model. *Advances in Science and Technology of Water Resources* **38** (2), 1–7.
- Wu, L., Wang, D. M. & Zhang, Y. 2006 Research on the algorithms of the flow direction determination in ditches extraction based on grid DEM. *Journal of Image and Graphics* **11** (7), 998–1003.
- Yao, C., Li, Z. J., Bao, H. J. & Yu, Z. B. 2009 Application of a developed grid-Xin'anjiang model to Chinese watersheds for flood forecasting purpose. *Journal of Hydrologic Engineering* **14** (9), 923–934.
- Yao, C., Li, Z. J., Zhang, K., Zhu, Y. L., Liu, Z. Y., Huang, Y. C., Gong, J. F., Zhang, J. T. & Tong, B. X. 2021 Fine-scale flood forecasting for small and medium-sized rivers based on grid-Xin'anjiang model. *Journal of Hohai University* **49** (01), 19–25.
- Ye, X. Y., Li, Q., Guo, Y. H., Liang, L. F. & Wang, Z. G. 2022 Progress of research on high-performance parallel distributed hydrological model. *Progress in Geography* **41** (4), 731–740.
- Zhang, A., Li, T., Si, Y., Liu, R., Shi, H., Li, X., Li, J. & Wu, X. 2016 Double-layer parallelization for hydrological model calibration on HPC systems. *Journal of Hydrology* **535**, 737–747.

First received 30 December 2022; accepted in revised form 7 March 2023. Available online 17 March 2023