

## **Parallel dynamic programming based on stage reconstruction and its application in reservoir operation**

**Huitao Zheng**

*State Key Laboratory of Water Resources and Hydropower Engineering Science, Wuhan University, Donghunan Road 8, Wuhan, Hubei 430072, China*

**Yadong Mei**

*State Key Laboratory of Water Resources and Hydropower Engineering Science, Wuhan University, Donghunan Road 8, Wuhan, Hubei 430072, China*  
*E-mail: ydmei@whu.edu.cn*

**Kai Duan**

*State Key Laboratory of Water Resources and Hydropower Engineering Science, Wuhan University, Donghunan Road 8, Wuhan, Hubei 430072, China*

**Yuru Lin**

*Yangtze River Scientific Research Institute, Huangpu Road 23, Wuhan, Hubei 430010, China*

### **Abstract**

We improved the serial recursion calculation process of dynamic programming and introduced parallel dynamic programming based on stage reconstruction. Through the proposed algorithm a multistage decision problem can be repeatedly reconstructed and gradually transferred to a single stage issue. This algorithm was applied to solve the optimized operation of cascade reservoirs in the lower reach of Yalong River in China. Results indicate that the calculating efficiency, compared with serial dynamic programming, can be significantly improved without sacrificing accuracy.

*Keywords:* parallel computing; optimized operation of cascade reservoir; parallel dynamic programming; stage reconstruction

### **1 Introduction**

Reservoir optimized operation is a group of complicated system-optimization problems, having features of high dimension, nonlinearity, and multiple constraints. Methods used to solve such problems fall into mainly two types: intelligent optimization algorithms and traditional optimization approaches. For intelligent optimization algorithms represented by genetic algorithm<sup>1, 2</sup>, as they are readily caught in locally optimal solution, their

application is constrained. As a traditional optimization method, dynamic programming<sup>3-6</sup> (DP) is the most classic and the most applied. However, with the number of reservoir power stations rise, the computation scale of DP increases exponentially and its computation timeliness drops sharply or even “curse of dimensionality” occurs. To improve the solving performance of intelligent algorithm, experts in China and around the world have conducted lots of research and made improvements on various algorithms<sup>7-10</sup>. However, as far as an intelligent

optimization algorithm is concerned, its search process is inevitably accompanied with certain randomness no matter which kind of improvement strategy is employed as long as it is under the frame structure of intelligent algorithm. Thereby, it cannot be guaranteed that the algorithm will converge at the globally optimal solution. For DP, scientists have proposed dimensionality reduction methods<sup>11, 12</sup> such as progressive optimal algorithm (POA) and discrete differential DP (DDDP). Although the complication of problem solving has been simplified by these dimensionality reduction methods, solving results more than often can only be regarded as satisfactory results when compared with results of standard DP as these methods rely heavily on initial solutions.

With the development and promotion of high performance computing technologies, the concept and computing technology of parallel are introduced to solve reservoir optimized operation. In Ref. 13, data decomposition plus the divide and rule mode were used to allocate computation tasks of different combinations of initial and final states within each time period to different CPU kernels for computation. In Ref. 14 and Ref. 15, the master process was used to designate and store DP computation tasks and results while slave process was used to compute, compare, and obtain the optimized accumulative objective function value. Essentially, master-slave mode was employed by these approaches in their progressive parallel treatment of state computation within a single stage. Improvement in the parallel performance of an algorithm mainly relies on the increase of CPU kernel (or process) number. In this study, on the basis of the recursion structure of standard DP, parallel DP (SR-PDP) based on stage reconstruction was introduced taking the stages of a multi-stage problem as the objects for parallel processing. The algorithm was implemented in VC++. And the effects of different controlling parameters on the computation performance of parallel algorithm were evaluated through analog computation using the cascade hydropower stations in the lower reach of the Yalong River as an example.

## 2 Mathematical model

Assuming that the initial and the final water levels of an operation period of each reservoir are known, under the condition of deterministic runoff, with the maximum

electric energy production as the goal, set up the optimization model of maximum electric energy production taking into account the constraint of cascade firm capacity.

The objective function is as follows

$$E = \max \sum_{t=1}^T \left[ N_t - \sigma_t A (NB - N_t)^\theta \right] \cdot \Delta t \quad (1)$$

$$N_t = \sum_{i=1}^n K_i \cdot Q_t^i \cdot H_t^i \quad (2)$$

where E is the total electric energy production of the cascade hydropower stations;  $N_t$  is the total capacity of the cascade hydropower stations at time period  $t$ ;  $NB$  is the firm capacity of the cascade hydropower stations;  $K_i$  is the capacity coefficient of the cascade hydropower station  $i$ ;  $Q_t^i$  is the average electric generation during time period  $t$  of hydropower station  $i$ ;  $H_t^i$  is the average electricity generation water head during time period  $t$  for hydropower station  $i$ ;  $\Delta t$  is the length of the computation time frame;  $\theta$  is an exponent, usually taking the value of 2;  $A$  is a positive penalty coefficient;  $\sigma_t$  is an variable with value of 0 or 1 and its value is decided by

$$\sigma_t = \begin{cases} 1 & N_t \leq NB \\ 0 & N_t > NB \end{cases} \quad (3)$$

The constraint conditions include:

(1) Constraint of flow connection between hydropower stations

$$QI_t^i = L_t^i + QX_t^{\Omega(i)} \quad (4)$$

where  $QI_t^i$  is the flow entering reservoir  $i$  in during time period  $t$ ;  $L_t^i$  is the local inflow of reservoir  $i$  during time period  $t$ ;  $\Omega(i)$  is an assembly of reservoirs at the upper reach and having direct flow connection with reservoir  $i$ ;  $QX_t^{\Omega(i)}$  is the total discharge from reservoirs belonging to  $\Omega(i)$  during time period  $t$ .

(2) Reservoir hydrologic budget constraint

$$V_t^i = V_{t-1}^i + (QI_t^i - QX_t^i) \cdot \Delta t \quad (5)$$

$$QX_t^i = Q_t^i + q_t^i \quad (6)$$

where  $V_{t-1}^i$  and  $V_t^i$  are the initial and final water storages, respectively, of reservoir  $i$  during time period  $t-1$ ;  $QX_t^i$  is the discharge of reservoir  $i$  during time period  $t$ ;  $q_t^i$  is the abandoned water flow of reservoir  $i$

during time period  $t$ .

(3) Hydropower station output constraint

$$N_{\min}^i \leq N_t^i \leq N_{\max}^i \quad (7)$$

where  $N_{\min}^i$  and  $N_{\max}^i$  are the lower limit and the upper limit, respectively, of the output from hydropower station  $i$ .

(4) Reservoir water storage amount constraint

$$V_{t,\min}^i \leq V_t^i \leq V_{t,\max}^i \quad (8)$$

where  $V_{t,\min}^i$  and  $V_{t,\max}^i$  are the lower limit and upper limit, respectively, of the water storage amount of hydropower station  $i$  during time period  $t$ .

(5) Reservoir discharged volume constraint

$$Q_{t,\min}^i \leq Q_t^i \leq Q_{t,\max}^i \quad (9)$$

where  $Q_{t,\min}^i$  and  $Q_{t,\max}^i$  are the lower limit and upper limit, respectively, of the discharged volume of hydropower station  $i$  during time period  $t$ .

(6) Boundary conditions

$$V_1^i = V_{\text{Begin}}^i \quad V_{T+1}^i = V_{\text{End}}^i \quad (10)$$

where  $V_{\text{Begin}}^i$  and  $V_{\text{End}}^i$  are the water storage amount at the beginning and the end of an operation period of reservoir  $i$ , respectively.

(7) Non-negative conditions constraint

All the above variables take non-negative values.

### 3 Serial DP

Taking the long term optimized operation of cascade hydropower reservoirs as an example. Assume that  $I$  is the number of hydropower reservoirs, an operation period is divided into  $T$  time periods,  $V_t$  is the water storage amount at the beginning of time period  $t$ ,  $QX_t$  is the discharged volume of time period  $t$ , and  $V_t = [V_t^1, \dots, V_t^I]^T$ ,  $QX_t = [QX_t^1, \dots, QX_t^I]^T$ . Under the condition that natural inflow is known, optimize operation strategies according to the optimization criterion that maximum electric energy production from hydropower reservoir groups is achieved during the operation period. The optimized operation is a typical multi-stage decision process. A common serial DP has the following computation steps:

(1) Stage variables: Divide the operation period into  $T$  stages based on month (or ten days) and use  $t$  to represent stage variable.

(2) State variables: The water storage amount  $V_t$  is selected as the state variable of a reservoir at time  $t$ .

(3) Decision variables: The discharged volume  $QX_t$  is selected as the decision variable for a reservoir for the time period  $t$ .

(4) State transfer function or hydrologic budget function:

$$V_{t+1} = V_t + (QI_t - QX_t) \cdot \Delta t \quad (11)$$

(5) Indicator function: If the electric energy production of a hydropower reservoir group at the  $t$  stage is selected as the indicator function, we have:

$$N_t = \psi(V_t, QX_t) \quad (12)$$

(6) The recursion relation:

$$f_t(V_t) = \underset{QX_t \in D_t(V_t)}{\text{opt}} \{ \psi(V_t, QX_t) + f_{t+1}(V_{t+1}) \} \quad (t = T, T-1, \dots, 1) \quad (13)$$

$$f_{T+1}(V_{T+1}) = 0 \quad (14)$$

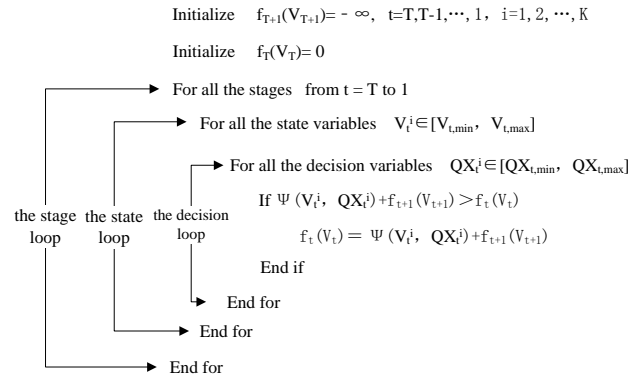


Fig. 1. The pseudo-codes for a serial DP recursive program

The program frame of serial DP recursion is mainly constituted by 3 nested major cycles which are the stage variable cycles, the state variable cycle, and the decision variable cycle. The pseudo-codes for the serial DP recursive program are shown in Fig.1.

Essentially, the stage cycle is the successive computation of the benefit to go  $f_{t+1}(V_{t+1})$ . Take the

backward computed serial DP as an example, the computation of  $f_i(V_i)$  relies on  $f_{t+1}(V_{t+1})$  from the next stage. Due to the interdependence between data in recursion, only sequential computation can be performed for standard DP.

**4 SR-PDP**

It can be seen from serial DP recursion function that each recursion in serial DP consists of 2 parts. The 1<sup>st</sup> part was the solving of a single stage indicator function; the 2<sup>nd</sup> part is the benefit to go after that stage until the last stage.

When solving DP using state discrete method, the solving order of the state points has no effects on the optimum-searching results of the current stage once the discrete point location of state space is confirmed. Therefore, employing the parallel computation technology, it is possible to segment state variables of DP and allocate them into different processes for parallel computation. In Ref. 14-16, essentially, master/slave mode was employed in the parallel treatment of the state combinatorial computing of a single stage. Therefore, it can be considered as a partial parallel of the indicator function  $\psi(V_t, QX_t)$ . However, the recursion method used is still that first adding the corresponding benefit to go to the indicator function value of the current stage and then carrying out recursion successively for each stage.

This study began from stage reconstruction for parallel computing. The basic idea is that dividing all the stages into several parallel subgroups and performing computation for each of the subgroups. Orderly connect the computation results of the above subgroups and result in a new multi-stage recursive problem which has fewer stages than the original problem. Performing such stage reconstruction repeatedly can progressively transfer a multi-stage problem into a single stage one, and the original problem is solved.

Taking into account the features of reservoir operation models and assuming that there are 12 stages during operation period and subgroup capacity (stages) is 2, perform uniform dispersion for the reservoir water level from its normal pool level to its dead water level. The solving process of SR-PDP is shown in Fig. 2 where the computing within each dashed box makes an independent subgroup.

For a multi-stage decision problem with T stages, if the subgroup capacity is C ( $C \geq 2$ ), the computation steps

are as follows:

- Step 1: Subgroups set up. For a multi-stage decision issue with T stages, from the first stage, every C stage makes a subgroup. Finally, a T-stage problem is divided into  $\lceil T/C \rceil$  independent subgroups. And the last subgroup has C or  $T - \lceil T/C \rceil * C$  stages;
- Step 2: Intersubgroup parallel computing. Intersubgroup parallel computing is simultaneously performed;
- Step 3: Intrasubgroup computing. Solve the C-stage subgroups according to standard DP and obtain results for different initial and end state combinations of the subgroups. Essentially, internal computing is a process that transfers a C-stage problem into a single stage one and records the corresponding local computation results of that subgroup;
- Step 4: Stage reconstruction. Orderly connect the previously obtained  $\lceil T/C \rceil$  single stage problems at their overlapping state points and reconstruct them into a new multi-stage problem. Now, the stage number changes from T to  $\lceil T/C \rceil$ ;
- Step 5: Judge whether it is the end. When the stage

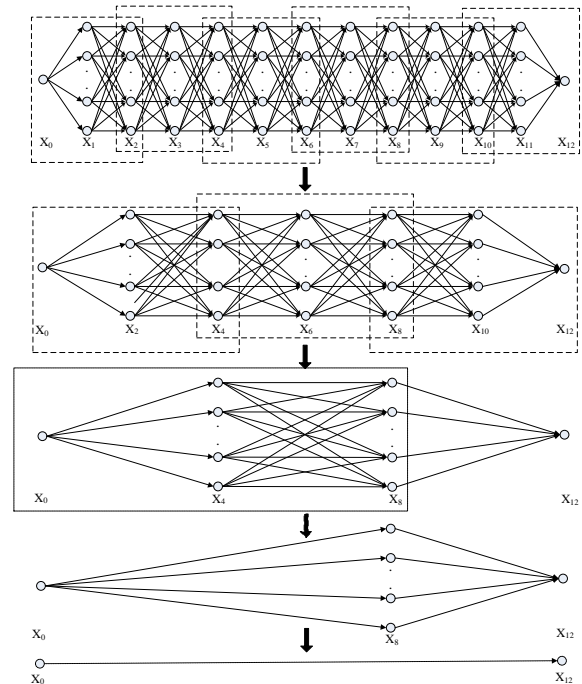


Fig. 2. The solving process of SR-PDP

### 5 Parallel performance evaluation

The major indexes that are used to evaluate the computation performance of a parallel algorithm are speedup ratio and parallel efficiency.

Speedup ratio is the speedup times used to evaluate a parallel machine cluster relative to serial machines. Suppose that it takes  $T_s$  for a certain serial application program to be run on a single CPU of a certain parallel machine. After parallelization of that program, it takes  $T_p$  for  $P$  programs to be run in parallel on  $P$  CPUs. Then, the speedup ratio  $S_p$  of that parallel program on that parallel machine can be defined as:

$$S_p = \frac{T_s}{T_p} \tag{15}$$

Parallel efficiency is used to evaluate the rate of the computing power that is effectively utilized. For an ideal parallel machine cluster, the speedup ratio is  $P$  and the parallel efficiency is 1. But, for a real system, the speedup ratio is less than  $P$  and the parallel efficiency is between 0 and 1.  $E$  stands for parallel efficiency and it describes what degree a CPU is effectively utilized. If communication overhead is not considered, parallel efficiency can be defined as:

$$E_p = \frac{S_p}{P} \tag{16}$$

### 6 Application examples

As shown in Fig. 3, the cascade hydropower stations at the lower reach of the Yalong River consists of 2 hydropower reservoirs that cannot be operated for a whole perfect year and 3 hydropower reservoirs with poor operation performance. Under the condition of deterministic runoff, with the maximum electric energy production as the goal, develop the optimization model of maximum electric energy production taking into account the confine of cascade firm capacity. As the 3 power stations including Jinping I, Jinping II and Gongdi Tongzilin were operated daily or of water diversion mode, in actual computing, the number of reservoirs that need operation strategy optimization was 2.

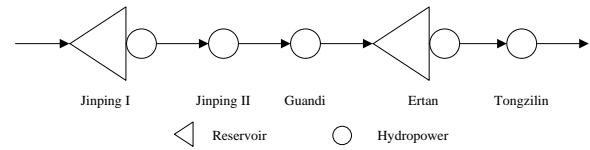


Fig. 3. Topological diagram of the cascade

The computing platform of the computer cluster was made up of 3 Intel® Core™ i5 computers of isostructuralism that were connected with each other. The CPU kernel number of the computer cluster was fixed as 12. The communication interface used MPICH2, and communication mode adopted the SPSD mode. Computing tasks were equally divided based on process numbers.

The effects of parameters such as subgroup capacity, state space discrete scheme and stage number in operation period on the computation performance of SR-PDP are discussed as follows taking the computation results of serial DP as standard.

#### 6.1 The effects of subgroup capacity

The 5-year measured data of monthly runoff from June 1959 to May 1964 were used for optimization computation. The stage number during the operation period was 60, and the state space was dispersed on 100 equal parts.

During the process of stage reconstruction, the processes needed by parallel computing became gradually fewer. As process number is related to subgroup capacity, the latter can exert great effects on the computation complexity and parallel efficiency of the algorithm. During computing, 6 (i.e., 5, 6, 7, 8, 9, and 10) subgroup capacity schemes were set up. Analog computation for the 6 schemes was performed using both serial DP and SR-PDP. Both computations arrived at a cascade electric energy production of  $4.05733 \times 10^{11}$  kW h and both cascade electricity generation assurance rates were 98.33%. These results confirmed that SR-PDP only changes the computation mode of serial DP without changing the final computation results of the algorithm.

Indexes of parallel computation performance for both serial and parallel computation modes are shown in Table 1. The number of processes needed in the repeated reconstruction computation of SR-PDP decreased

progressively with the increase in the times of reconstruction. In other words, process number was not a fixed value. Therefore, discussion on the parallel computation performance of SR-PDP here did consider the index of parallel efficiency. Instead, the parallel computation time and parallel speedup ratio were employed for the evaluation of parallel performance of the algorithm.

As can be seen from Table 1, with the increase of subgroup capacity, the parallel computation performance of SR-PDP showed trends of first going up and then dropping, which is shown in Fig. 4.

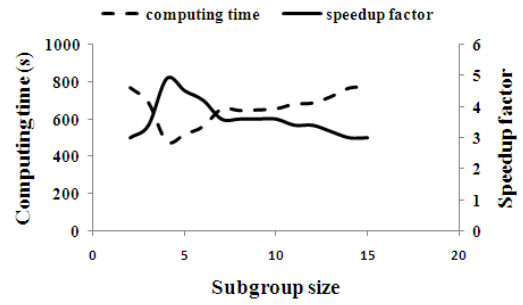


Fig. 4. Relationship between the parallel computation performance of SR-PDP and subgroup capacity

Table 1 Comparison of computation results of different subgroup capacity schemes

Scheme No.	Subgroup capacity	Changing of process number	Computation scale (10 <sup>8</sup> times)	Computation time (s)		Speedup ratio
				Serial	Parallel	
SR-PDP-A1	2	30→15→8→4→2→1	ca 58		771	3.0
SR-PDP-A2	3	20→7→3→1		2341	690	3.4
SR-PDP-A3	4	15→4→1		482	4.9	
SR-PDP-A4	5	12→3→1		522	4.5	
SR-PDP-A5	6	10→2→1		563	4.2	
SR-PDP-A6	7	9→2→1		657	3.6	
SR-PDP-A7	8	8→1		648	3.6	
SR-PDP-A8	9	7→1		652	3.6	
SR-PDP-A9	10	6→1		658	3.6	
SR-PDP-A10	11	6→1		683	3.4	
SR-PDP-A11	12	5→1		689	3.4	
SR-PDP-A12	13	5→1		723	3.2	
SR-PDP-A13	14	5→1		769	3.0	
SR-PDP-A14	15	4→1		775	3.0	

With the set up of the parameters in this section, when subgroup capacity was 4-6, computation time was the shortest and parallel computation performance fell in the optimal range. When process number was equal to or slightly larger than the CPU kernel number of the computation platform, the computation platform set up in this study can achieve its highest parallel computation performance. As the computation platform was a simple

computer cluster of 3 4-kernel PC machines interconnected through local area network and the CPU kernel number was 12, it was better for the value of the subgroup capacity C to be a positive integer close to  $\lceil T/12 \rceil$  to avoid too many processes which will lead to a drop in parallel computation performance.

### 6.2 The dispersion number of state space

The measured monthly runoff data of 1 year from June

1959 to May 1960 were used for optimization computation. The stage number of the operation period was 12, and the subgroup capacity was fixed as 2. Ten schemes of state space dispersion number were set up from 10 to 100 with a step size of 10.

Analog computations were performed for the above 10 schemes using both the serial DP and parallel DP. Optimization results are shown in Table 2.

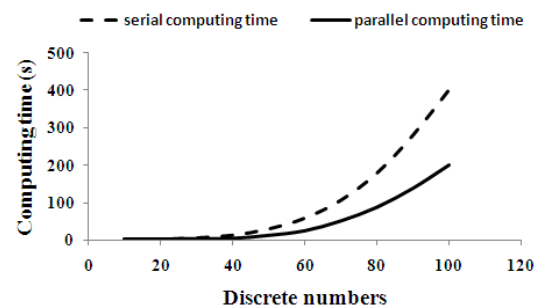
As can be seen from Table 2, with the dispersion number of state space increased, the electricity generation displayed increasing trends and partial vibration occurred during the increasing process. The reason is that uniformly divided fixed dispersion over water level was used in this study. Changes of the water level dispersion point location might have resulted in the slight bias in electric energy production.

Table 2 Comparison of analog computation results of different dispersion schemes

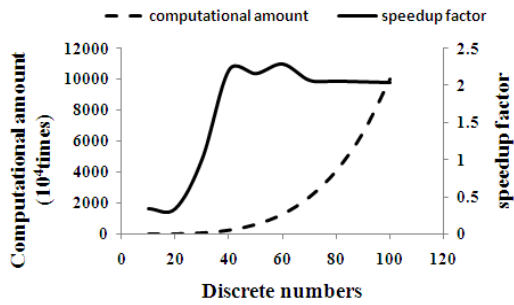
Scheme No.	Dispersion number	Electric energy production (10 <sup>8</sup> kW h)	Cascade electricity generation assurance rate (%)	Computation scale (10 <sup>4</sup> times)	Computation time (s)		Speedup ratio
					Serial	Parallel	
SR-PDP-B1	10	715.47	91.67	ca 1.02	1	3	0.33
SR-PDP-B2	20	719.389	91.67	ca 16.08	1	3	0.33
SR-PDP-B3	30	719.97	100.00	ca 81.18	4	4	1.00
SR-PDP-B4	40	719.758	100.00	ca 256.32	11	5	2.20
SR-PDP-B5	50	719.913	100.00	ca 625.50	28	13	2.15
SR-PDP-B6	60	719.939	100.00	ca 1296.72	57	25	2.28
SR-PDP-B7	70	720.33	100.00	ca 2401.98	105	51	2.06
SR-PDP-B8	80	719.879	100.00	ca 4097.28	178	87	2.05
SR-PDP-B9	90	720.317	100.00	ca 6562.62	280	137	2.04
SR-PDP-B10	100	720.472	100.00	ca 10002.00	402	198	2.03

Computation scale and computation time increased rapidly in response to the increase of the dispersion number of state space. After comparison, we can see from Fig. 4 that the computation time of SR-PDP decreased fairly greatly compared with that of serial DP. When the dispersion number was between 10 and 30, serial computation time was relatively short, leading to fairly big changes in the parallel computation performance of SR-PDP. When the dispersion number was over 30, the parallel computation performance of SR-PDP fluctuated slightly with the increase of the dispersion number of state space, but the fluctuation was negligible when compared with the total computation time. Therefore, it can be concluded that when other parameters were fixed and the problem to be solved required computation amount at scale, the computation

performance of SR-PDP basically stayed stable as shown in Fig. 5.



(a) Comparison of computation time



(b) The relationship between speedup ratio and computation scale

Fig. 5. Comparison of analog results

### 6.3 Stage number in an operation period

The dispersion number of state space was fixed as 100, and the subgroup capacity was fixed as 5. Thirty schemes of stage number were set up from 1 to 30 years with 1 year as step size.

Analog computation was carried out for the above 30 schemes using both the serial DP and parallel DP. The optimization results are shown in Table 3.

As can be seen from Table 3, the cascade electric energy production assurance rates were all above 95% for all the different stage number schemes, which met the requirement of actual projects. Therefore, the computation results of parallel DP were valid for actual projects. With the increase of stage number, roughly, the computation scale increased linearly, and the computation time of serial DP also exhibited trends of linear increasing with the increase in computation scale. The computation time of SR-PDP dropped fairly greatly compared with that of serial DP as shown in Fig. 6. With the increase of stage number, the computation time of

SR-PDP tended to increase stairwise as shown in Fig. 7.

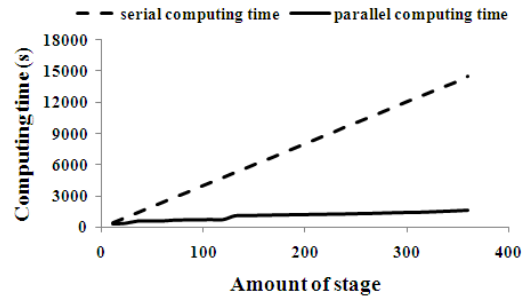


Fig. 6 Comparison of computation time

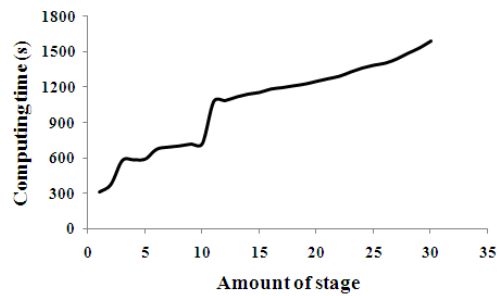


Fig. 7 The relationship between computation time and stage number

The reason for the stairwise increase of computation time with increasing stage number is that the subgroup capacity was fixed as 5, and the stage reconstruction times needed during the computation process would increase stairwise with the increase in stage number as shown in Table 4.

Table 4 The relationship between stage reconstruction times and stage number

Stage number	12	24	36	48	60	72	84	96	108	120
Reconstruction times	2	2	3	3	3	3	3	3	3	3
Stage number	132	144	156	168	180	192	204	216	228	240
Reconstruction times	4	4	4	4	4	4	4	4	4	4
Stage number	252	264	276	288	300	312	324	336	348	360
Reconstruction times	4	4	4	4	4	4	4	4	4	4



Table 3 Comparison of analog results of schemes with different stage numbers

Scheme No.	Stage number	Cascade electric energy production ( $10^8$ kW h)	Cascade electricity generation assurance rate (%)	Computation scale ( $10^8$ times)	Computation time (s)		Speedup ratio
					Serial	Parallel	
SR-PDP-C1	12	720.47	100.00	ca 10	402	305	1.32
SR-PDP-C2	24	1534.29	95.83	ca 22	889	370	2.40
SR-PDP-C3	36	2428.74	97.22	ca 34	1371	573	2.39
SR-PDP-C4	48	3290.31	97.92	ca 46	1863	579	3.22
SR-PDP-C5	60	4057.33	98.33	ca 58	2341	585	4.00
SR-PDP-C6	72	4936.63	98.61	ca 70	2834	669	4.24
SR-PDP-C7	84	5747.81	98.81	ca 82	3300	687	4.80
SR-PDP-C8	96	6603.68	97.92	ca 94	3806	697	5.46
SR-PDP-C9	108	7311.31	98.15	ca 106	4278	713	6.00
SR-PDP-C10	120	8083.09	98.33	ca 118	4755	717	6.63
SR-PDP-C11	132	8972.87	98.48	ca 130	5249	1076	4.88
SR-PDP-C12	144	9693.36	98.61	ca 142	5721	1083	5.28
SR-PDP-C13	156	10487.1	98.72	ca 154	6217	1114	5.58
SR-PDP-C14	168	11267.7	98.81	ca 166	6670	1138	5.86
SR-PDP-C15	180	12105.3	99.44	ca 178	7170	1152	6.22
SR-PDP-C16	192	12954.6	99.48	ca 190	7651	1181	6.48
SR-PDP-C17	204	13770.7	99.51	ca 202	8134	1193	6.82
SR-PDP-C18	216	14580.4	99.54	ca 214	8627	1209	7.14
SR-PDP-C19	228	15276.8	99.56	ca 226	9097	1224	7.43
SR-PDP-C20	240	16037.7	99.58	ca 238	9618	1247	7.71
SR-PDP-C21	252	16868.4	99.60	ca 250	10079	1269	7.94
SR-PDP-C22	264	17612.1	99.62	ca 262	10549	1290	8.18
SR-PDP-C23	276	18454.7	99.64	ca 274	11037	1327	8.32
SR-PDP-C24	288	19325.6	99.65	ca 286	11529	1360	8.48
SR-PDP-C25	300	20164.4	99.33	ca 298	12032	1384	8.69
SR-PDP-C26	312	20988.5	99.36	ca 310	12487	1401	8.91
SR-PDP-C27	324	21871.4	99.38	ca 322	12974	1437	9.03
SR-PDP-C28	336	22640.1	99.40	ca 334	13445	1486	9.05
SR-PDP-C29	348	23489.9	99.43	ca 346	13907	1531	9.08
SR-PDP-C30	360	24186.5	99.44	ca 358	14472	1589	9.11

The computation scale increased linearly with the increase of stage number. The increase of parallel speedup ratio of SR-PDP is shown in Fig. 8. When the stage number was 132, the speedup ratio displayed decrease and then continued increasing. This was because when the subgroup capacity was 5 and the stage number increased from 120 to 132, the stage reconstruction times needed changed from 3 to 4. Thereby, the computation time showed fairly great increase, leading to the big decrease of the speedup ratio of SR-PDP here.

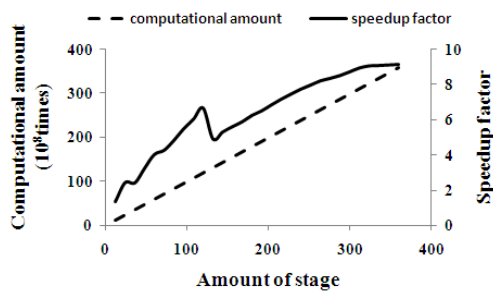


Fig. 8 The relationship between computation scale, speedup ratio, and stage number

## 7 Conclusions

The following conclusions are drawn based on the simulation and analysis of the computation performance of SR-PDP:

(1) Compared with DP, SR-PDP can greatly shorten the computation time without sacrificing the computational accuracy.

(2) For different subgroup capacity schemes, the computation time of SR-PDP was shortened fairly greatly compared with that of serial DP. The changing trend of the parallel performance of SR-PDP was related to the relationship between subgroup capacity and the total stage number of an operation period. When the subgroup capacity was so set up that the process number needed for computation was smaller or slight greater than the CPU kernel number of the computation platform, the parallel performance of SR-PDP was in direct proportion to the subgroup capacity. When the subgroup capacity was so set up that the process number needed for computation was far greater than the CPU kernel number of the

computation platform, the parallel performance of SR-PDP was inversely proportional to the subgroup capacity.

(3) For different dispersion schemes of state space, the computation time of SR-PDP decreased fairly greatly compared with that of serial DP. Generally, as long as the computation volume of a problem reached a certain scale, the parallel computation performance of SR-PDP basically stayed stable with the changes of the dispersion number of state space.

(4) For different schemes of stage number of an operation period, the computation time of SR-PDP was shortened fairly greatly compared with that of serial DP. With the increase of the stage number in an operation period, the parallel computation performance of SR-PDP tended to increase stairwise. The location of the stage jump point was related to the relationship between the subgroup capacity and the stage number of an operation period. Generally speaking, when the subgroup capacity is so set up that the reconstruction times of SR-PDP need to be raised, jump of computation performance will occur in which the subgroup capacity is and the computation performance of the subgroup capacity schemes at locations slightly after the jump point will be lower than that of the schemes slightly before the jump point.

## Acknowledgements

This work was supported by the Major Program of National Natural Science Foundation of China (51239004) and the General Program of National Natural Science Foundation of China (50779049).

## References

1. T. Jiang, N. Liang, L. Kang and D. Huang, Automatic generation schedules control by using genetic algorithm, *Journal of hydroelectric engineering* **4** (1995) 7-14. (in Chinese)
2. G. Ma and L. Wang, Application of a genetic algorithm to optimal operation of hydropower station, *Advances in water science* **8**(3) (1997) 275-280. (in Chinese)
3. C. Ji and S. Feng, Optimization of energy indexes and long-term operation of multireservoir power systems (application of discrete differential dynamic programming), *Journal of Wuhan university of hydraulic and electric engineering* **3** (1984) 87-95. (in Chinese)
4. Y. Mei, Dynamic programming model and method of cascade reservoirs optimal operation for flood control, *Journal of Wuhan university of hydraulic and electric*

- engineering* **32**(5) (1999) 10-12. (in Chinese)
5. K. D. W. Nandalal and J. J. Bogardi, *Dynamic programming based operation of reservoirs: applicability and limits* (Cambridge University Press, New York, 2007).
  6. Y. Mei, Y. Xiong and L. Chen, A dynamic programming method for the multi-purpose operation of cascade reservoirs, *Journal of hydroelectric engineering* **26**(2) (2007) 1-4. (in Chinese)
  7. X. Wan, J. Zhou, Application of genetic algorithm for self-adaptation, symmetry and congruity in reservoir mid-long hydraulic power operation, *Advances in water science* **18**(4) (2007) 598-603. (in Chinese)
  8. J. Zheng, K. Yang, F. Ni and G. Liu, Research on overall improved genetic algorithm applied in optimal generation dispatching of multi-reservoir system, *Journal of hydraulic engineering* **44**(2) (2013) 205-211. (in Chinese)
  9. D. N. Kumar and M. J. Reddy, Multipurpose reservoir operation using particle swarm optimization, *Journal of Water Resources Planning and Management* **133**(3) (2007) 192-201. (in Chinese)
  10. M. J. Reddy and D. N. Kumar, Evolving strategies for crop planning and operation of irrigation reservoir system using multi-objective differential evolution, *Irrig. Sci.* **26** (2008) 177-190.
  11. H. Zong, C. Li, J. Zhou and Y. Zhang, Research and application for short-time cascade hydroelectric scheduling based on progressive optimality algorithm, *Water Resources and Power* **21**(1) (2003) 46-48. (in Chinese)
  12. Z. Dong, *The optimization theory and application in scheduling and planning reservoirs*. (Shandong Science and Technology Pres, Jinan, 1989). (in Chinese)
  13. C. Cheng, X. Gao, X. Wu and S. Gao, Fine-grained parallel discrete differentiation and dynamic programming algorithm for long-term optimization of cascade hydropower system, *Proceedings of the CSEE* **31**(10) (2011) 26-32. (in Chinese)
  14. X. Wan and G. Wang, Parallel dynamic programming for hydropower generation optimization, *Journal of Hydroelectric Engineering* **30**(6) (2011) 166-170. (in Chinese)
  15. X. Li, J. Wei and C. Yao, Parallel dynamic programming for multi-reservoir system optimization, *Journal of Tsinghua University (Sci&Technol)* **53**(9) (2013) 1235-1240. (in Chinese)