

并行动态规划和改进遗传算法在水库调度中的应用

张忠波^{1,3}, 吴学春², 张双虎³, 蒋云钟³, 耿思敏³, 王浩^{1,3}

(1. 天津大学水利工程仿真与安全国家重点实验室, 天津 300072;

2. 南水北调东线江苏水源有限责任公司, 南京 210029;

3. 中国水利水电科学研究院流域水循环模拟与调控国家重点实验室, 北京 100038)

摘要: 多核处理器的日益普及为并行计算提供了条件, 将并行计算与动态规划相结合, 用 OpenMP 编程模式实现动态规划的并行计算, 可大大提高计算效率。采用遗传算法求解优化问题, 初始种群优劣直接影响到求解效率, 采用广度变异和分层进化模式生成初始种群, 可有效改进初始种群的鲁棒性, 提高了遗传算法求解效率。将上述两种优化算法应用于水库优化调度模型求解中, 计算结果表明: 该方法易于实现, 求解效率高, 为水库优化调度模型求解提供了新的途径。

关键词: 水文学; 水库调度; 并行动态规划; OpenMP; 分层遗传算法

中图分类号: TV697.1⁺1 文献标识码: A

Parallel dynamic programming and improved genetic algorithm and their application to reservoir operation

ZHANG Zhongbo^{1,3}, WU Xuechun², ZHANG Shuanghu³, JIANG Yunzhong³, GENG Simin³, WANG Hao^{1,3}

(1. State Key Laboratory of Hydraulic Engineering Simulation and Safety, Tianjin University, Tianjin 300072;

2. Jiangsu Water Resources Co., Ltd. of South to North Water Transfer Eastern Route Project, Nanjing 210029;

3. State Key Laboratory of Simulation and Regulation of Water Cycle in River Basin, China Institute of Water Resources and Hydropower Research, Beijing 100038)

Abstract: The increasing popularity of multicore processors provides a condition for parallel computation, and a combination of parallel computation with dynamic programming implemented by an OpenMP model can raise computational efficiency greatly. In application of a genetic algorithm to optimization problem, its solution efficiency depends on initial population, but its robustness and efficiency can be improved through generation of initial population with a breadth mutation model or a hierarchy evolution model. We have applied these two models separately to solution of reservoir optimal operation. Results show that they are effective and easy to implement and would provide new approaches to solution of reservoir optimal operation model.

Key words: hycerology; reservoir operation; parallel dynamic programming algorithm; OpenMP; hierarchy genetic algorithm

0 引言

水库优化调度是典型的多维非线性优化问题, 求解方法显得尤其重要。动态规划(DP)是由 Bellman(1957)提出的用于解决多阶段决策过程最优化问题的一种数学方法^[1]。它是最优化领域中一个重要分支, 是一种研究多段决策过程的递推最优化方法。它可以将复杂的初始问题划分为若干个阶段的子问题, 逐时段求解, 而水库调度正是一种与时间过程相关的典型动态多阶段决策过程, 决策具有无后效性, 也是水库调度中应用最多的方法之一。但是采用动态规划等传统算法^[2]在实际应用中常常会遇到“维数灾”的问题, 由于动态规划法求解时, 随着计算精度的提高, 计算所需时间成指数倍增长, 计算效率明显降低。人

收稿日期: 2012-11-09

基金项目: 国家重点基础研究发展计划(973 计划)资助(2013CB036406)

作者简介: 张忠波(1986-), 男, 博士研究生. E-mail: zzbaiowen@126.com

通信作者: 吴学春(1969-), 男, 高级工程师. E-mail: wuxc136@163.com

们又提出了不少改进的传统优化算法,但在某些方面仍有其缺陷。如逐次逼近动态规划难以保证收敛到全局最优解等;毛睿^[3]等提出了基于并行分布式计算方法进行库群优化调度,可大大降低计算时间。

针对水库调度优化问题,人们在实践中提出生物进化算法。遗传算法(Genetic Algorithm)最初由美国Michigan大学J.Holland教授于1975年首先提出来的,但是经典的遗传算法不是完全遍历的马尔可夫过程,收敛速度慢、接近全局最优解时很难收敛,且容易早熟。特别对于复杂非线性问题(水库优化调度),更容易发生局部收敛。许多学者从编码方式的角度改进遗传算法,马光文等人^[4]采用二进制编码的遗传算法在水库优化调度中得到应用;畅建霞等人^[5]采用了十进制编码的优化方法改进遗传算法;陈立华和梅亚东等人^[6]采用超立方体浮点编码遗传算法对遗传算法有一定的改进。采用自适应的交叉率和变异率可以改进遗传算法的收敛性^[7~8];但是在交叉和变异过程中随机性较大,随机初始种群多样性较差,随着种群适应度提高而局部收敛,初始种群优劣一定程度上决定了算法全局收敛性。

水库优化调度从基本的动态规划算法和遗传算法两种求解方法出发,并在动态规划算法加入并行模块,提高计算效率,在一定程度上缓解了“维数灾”;采用分层自适应遗传算法,改善了初始种群的多样性,提高了收敛速度,改善了调度效益;结合三峡水库实际调度,并采用改进的算法求解水库优化调度模型。

1 水库调度模型

1.1 目标函数

对于水库优化调度决策,一般来讲都是多目标决策问题,目标包括防洪、发电、航运以及下游用水效益等。如果单以发电量为目标函数可以写成:

$$F = \max \sum_{t=1}^T KQ_t H_t T_t \text{ 或 } F = \max \sum_{t=1}^T N_t T_t \quad (1)$$

式中: F 为电站年发电量, kW·h; K 为电站综合出力系数,无量纲; Q_t 为电站在 t 时段发电流量, m^3/s ; H_t 为电站在 t 时段平均发电净水头, m; T 为年内计算总时段; N_t 为时段的出力, kW; T_t 为时段时间, s。

在实现算法优化过程中,一般把算法的适应度转换成带有约束条件的单目标。在水库优化调度中适应度函数通常采用惩罚函数法。采用定量惩罚函数法,通过加大惩罚系数的方法来抑制不可行解的产生。如式(2)。

$$f(V_t) = F(V_t) + \alpha \min\{N - N_{\min}, 0\} T_t \quad (2)$$

式中: $f(V_t)$ 为时段 t 适应度; $F(V_t)$ 为时段 t 发电量; α 为惩罚系数,无量纲。

1.2 约束条件

① 水量平衡约束 $V_{i+1} = V_i + (I_i - O_i) \Delta t$, 式中, V_i 是 i 时段的水库库容, m^3 ; I_i 是 i 时段的入库流量, m^3/s ; O_i 是 i 时段的出库流量, m^3/s ; Δt 是时段长度, s;

② 水位约束: $Z_{\min} \leq Z_t \leq Z_{\max}$, Z_{\min} 、 Z_t 、 Z_{\max} 对应最低水位、时段 t 水位和最高水位, m;

③ 库容约束: $V_i^l \leq V_i \leq V_i^u$ ($i=1, 2, \dots, T$), V_i^l 是 i 时段的最小库容, m^3 ; V_i^u 是 i 时段的最大库容, m^3 ;

④ 水库出库流量约束 $O_l \leq O_t \leq O_u$, O_l 是水库的最小出库流量, m^3/s ; O_u 是最大泄流流量, m^3/s ;

⑤ 出力约束 $N_{\min} \leq N_t \leq N_{\max}$, N_{\min} 、 N_t 、 N_{\max} 对应最小出力、 t 时段出力和装机容量, kW。

2 算法的改进

2.1 动态规划算法应用

1) 基本的动态规划算法

适用动态规划的问题必须满足最优化原理和无后效性。动态规划法适用于求解水库优化调度模型,具体求解步骤如下:

① 阶段变量: 由于水库优化调度是按照时间过程进行的,属于多段决策过程,阶段变量按时间段选取。

② 状态变量: 状态变量选取水库对应变量,例如水位、库容等。其水库水位和库容反应了调度过程的演变,并满足无后效性要求。

③ 决策变量: 时段末的水库状态对应一个决策,即决策变量。

④ 状态转移规律：根据决策变量从而得到时段末的一个状态，作为下一时段的起始状态。

⑤ 递推方程： $f(V_t) = \max \{ f(V_t, Q_t) + f(V_{t+1}) \}$ $t = 1, 2, \dots$ 式中， $f(V_t, Q_t)$ 为面临时段 t 通过决策得到面临时段的目标适应值； $f(V_t)$ 为从时段 t 到调度期末的最优目标适应值； $f(V_{t+1})$ 为时段 t 后最优目标适应值。

2) 并行算法设计

水库优化调度模型的求解过程具有一定的并行性，有利于并行计算的实施。多核处理器的日益普及为并行机制的实现提供了必要的硬件基础。为了防止多线程计算而导致计算资源的浪费；在动态规划算法中嵌入并行模块。并行计算 (Parallel Computing) 是指，将一个目标问题分解成多个子任务，分配给不同处理结构，各个结构之间相互协同，在同一时段内，并行地执行子任务，从而达到加速求解速度，或者减少目标问题的规模程度。OpenMP 采用共享存储的编程模式，所谓共享存储，是指各个处理器^[8]可以对共享存储器中的数据进行存取，数据对每个处理器而言都是可以访问到的，不需要在处理器之间进行传递，即数据通信是通过读或写共享存储单元来完成。

OpenMP 是基于线程的并行编程模型，并采用了 Fork-Join 的并行模型。所有 OpenMP 程序开始于一个主线程 (Master Thread)，并一直串行执行，直到遇见第一个并行域 (Parallel Region) 才开始并行。即①Fork：主线程创建一行并行线程，并行域中代码在不同线程队中运行；②Join：当所有的并行域执行完之后，它们或被同步或被中断，最后回到主线程的运行。所有的 OpenMP 的并行化，都是通过嵌入到源代码中的编译制导语句来实现并行化的，如图 1。

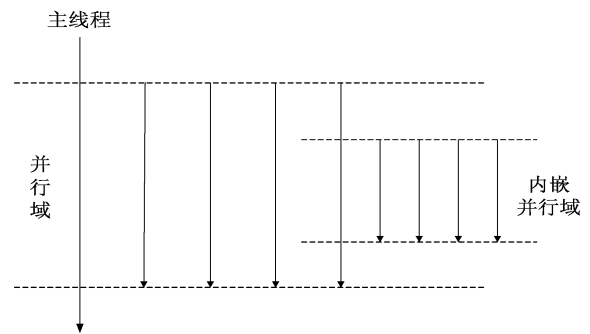


图 1 Fork-Join 执行模式
Fig.1 Mode of Fork-Join running

根据复杂度理论，衡量算法效率指标是计算任务所使用的时间和空间资源。在水库调度动态规划计算中，计算量最大的部分就是重复计算决策变量对应的适应度值。假设电站水位离散个数为 k ，除了起始和终止蓄水状态，其余阶段始末各有 k 个组合状态，需进行 k^2 次计算。若时段数为 T ，每轮寻优需要 $2k+k^2(T-2)$ 次计算，时间复杂度为 $O(k^2T)$ ，所以比较容易遭遇“维数灾”。加入并行设计后，假设 CPU 核数为 P ，并行后会分成 P 个线程同时计算，则每轮迭代时需要计算 $[2k+k^2(T-2)]/P$ 次。由于线程与主线程之间的通信是在临界区，效率较高^[9]，因此并行后复杂度应为 $O((k^2T/P)+P)$ 。可见并行算法的设计能够在相同的时间内能够完成更多的工作。

3) 动态规划实现并行化的具体步骤：

① 在并行域前，首先主线程计算水库的各个时段的所有的决策方案，并用并行化嵌套语句设置多个子线程。

② 每个子线程从对应共享内存中获取计算数据，包括电站的基础属性数据、各类特征曲线和约束条件等。每一个线程由一个 CPU 内核处理，同时每一个子线程只允许与主内存进行通信和数据交互，核与核是相互隔离的，从而保证各个线程独立工作。

③ 子线程通过与共享内存的通信得到数据，计算当前时段的一个决策方案的发电量和目标适应值，并将结果返回到主内存中，从而达到了汇合。

④ 当所有子线程执行完成后，结束并行计算，回到主线程，回到主线程环境中。

2.2 改进遗传算法应用

遗传算法是模拟达尔文生物进化论的自然选择和遗传学机理的生物进化过程的计算模型，是一种通过模拟自然进化过程搜索最优解的方法，通过种群交叉变异进化寻优的启发式的搜索算法。在自适应遗传算法寻优过程中，用适应度函数评价每一代个体的优劣，通过对整个参数空间编码得到待处理种群，再对其检测并选择优良个体进行随机交叉得到下一代种群，新一代种群中优良个体的性状得以保留，同时适应度低的个体将被淘汰。

改进的自适应遗传算法分两层，第一层采用无搜索策略的广度搜索算法寻找种群的多样性，通过外部

存档的方式储存精英个体^[10]，增加种群多样性，从而增强起初母体的鲁棒性；第二层采用第一层外部存档的初始种群，并嵌套广度搜索的变异模块，通过合理判定条件进入随机变异，防止陷入局部最优，并采用自适应遗传算法进行全局搜索。分层收敛算法具体设计步骤如下：

1) 第一层：

① 初始化种群 pop ；

② 计算种群目标函数 m ；为了保证算法在进化过程中种群的多样性，在进入自适应遗传算法模块之前首先进行“广度”变异，所谓“广度”是相对于具有自适应遗传算法的变异率和交叉率而言，依靠随机数在“精英”个体周围变异，如式 (3)：

$$\begin{cases} (V_1^i, \dots, V_n^i) = (V_1^{best}, \dots, V_n^{best}) - r \times ((V_1^{best}, \dots, V_n^{best}) - (V_1^{min}, \dots, V_n^{min})) & i \leq pop/2 \\ (V_1^i, \dots, V_n^i) = (V_1^{best}, \dots, V_n^{best}) + r \times ((V_1^{max}, \dots, V_n^{max}) - (V_1^{best}, \dots, V_n^{best})) & i > pop/2 \end{cases} \quad (3)$$

采用不具备搜索策略的广度变异，文献[6]中采用保留全局最优个体直接进入下一代方式，保证最优个体不被破坏。为了保证算法搜索广度，确保种群的多样性，不采用保留最优个体措施。通过外部存档策略将每一代最优个体存储起来，得到第二层对应的初始种群。

2) 第二层：

① 采用第一层得到的具有较好多样性外部存档作为初始种群 pop ；

② 计算种群目标函数 m ；

③ 根据种群函数适应值 $fitness$ ，以交叉概率 P_c 和变异概率 P_m (如式 (5)，(6)) 进行交叉、变异和精英选择策略操作^[7]，第二层同样为了抑制算法在局部最优收敛强制嵌入“广度变异”模块，并根据式 (4) 判定是否进入嵌套的广度变异模块，得到新一代种群：

$$|f_{max}^i - f_{max}^{i-1}| \leq \varepsilon \quad (4)$$

$$P_c = \begin{cases} P_{c1} - \frac{(P_{c1} - P_{c2})(f' - f_{avg})}{(f_{max} - f_{avg})} & f' \geq f_{avg} \\ P_{c1} & f' < f_{avg} \end{cases} \quad (5)$$

$$P_m = \begin{cases} P_{m1} - \frac{(P_{m1} - P_{m2})(f' - f_{avg})}{(f_{max} - f_{avg})} & f' \geq f_{avg} \\ P_{m1} & f' < f_{avg} \end{cases} \quad (6)$$

式中： f_{max} 为群体中最大适应度值； f_{avg} 为每代群体平均适应度值； f' 为要交叉的两个个体中较大适应度值； f 为要变异个体的适应度值； $P_{c1}=0.9$ ， $P_{c2}=0.6$ ， $P_{m1}=0.1$ ， $P_{m2}=0.001$ 。

④ 收敛性判断：检验是否达到进化代数，若达到，输出最优解，否则转向步骤②。分层进化算法的步骤示意图如图 2。

为了证明分层遗传算法的改进效果，同时又设计了另外两个算法：

① 自适应遗传算法

单纯地采用自适应的遗传算法求解水库优化问题，首先根据公式 (7) 进行个体编码，然后根据遗传算子进行迭代计算，采用式 (5)、(6) 的交叉率和变异率的自适应遗传策略迭代寻优，判定标准采用第二层的步骤④。

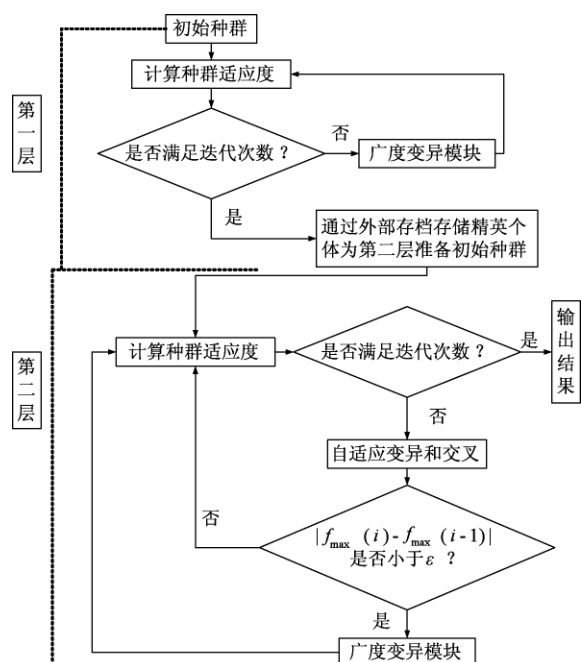


图 2 分层算法的步骤示意图
Fig.2 Flowchart of the hierarchy algorithm

② 广度变异算法

在自适应遗传算法的基础上，并加入了第二层的步骤③，其它设计与自适应遗传算法相同。

3 实例应用

三峡工程是治理开发长江的关键性工程，水库正常蓄水位 175m，防洪限制水位 145m，枯期消落低水位 155m，具有巨大的防洪、发电、航运和枯期向下游补水等综合利用效益，三峡水库防洪库容 221.5 亿 m³，兴利库容 165.0 亿 m³，有巨大的调蓄能力。

由于三峡水库可进行年调节，而汛期基本保持汛限水位不变，所以对供水期进行优化调度更加合理，即从开始蓄水到翌年的汛前进行优化调度，采用旬为计算时段。以典型枯水年供水期作为研究对象，选取三峡坝址宜昌水文站的天然来水 95% 保证率的 1972 年的径流系列作为研究依据。

3.1 并行动态规划算法结果分析

采用设计的并行动态规划算法进行优化求解，求解确定型动态规划模型时，结合 Fork-Join 并行框架，并通过 Fortran 编程在多核 CPU 环境下实现，所采用主机类型为 lenovo 机架，CPU 类型为 Intel Quad Core 4 核。为了进一步验证并行计算的效率，在程序中设置了不同的线程，计算时间如表 1。

表 1 串并行计算参数
Table 1 Parameters of serial and parallel computations

参数	序号	串行	并行（不同线程）			
			2	4	8	16
计算时间/s	1	27.79	15.09	7.98	8.96	8.45
	2	27.79	15.09	7.98	8.78	9.06
	3	27.79	15.09	7.98	9.06	8.54
平均时间/s	1	27.79	15.09	7.98	8.93	8.68
CPU 使用率/(%)	1	25	50	100	100	100

从表 1 可以看出，并行计算要比串行计算时间明显要少；并行中不同线程的计算时间也不同，计算时间并不是随着线程的增加不断减少，对于本台一个 CPU 四个核心的计算机而言，4 个线程的计算时间最小。同时串行计算 CPU 的使用率仅为 25%，而双线程达到了 50%，4、8、16 线程 CPU 使用率达到了 100%；可知串行计算未充分利用计算机资源，相反并行计算中 CPU 使用率达到了 100%，充分利用了计算机资源。

3.2 改进遗传算法应用结果分析

改进算法的目标函数采用优化调度模型中公式 (1)，其适应度采用与传统优化算法相同的适应度公式 (2)。遗传算法的前处理是个体编码，文献[7]中采用水位的实数编码，而对于大型水库，在较高水位情况下，同等水位的变化会引起对应库容变化很大，在算法搜索中产生的子代的遍历性不够理想。因此，研究采用离散库容的方法进行实数编码。个体的基因对应水库库容实际值。

$$V_t = V_{t,\min} + r(V_{t,\max} - V_{t,\min}) \tag{7}$$

式中： $V_{t,\min}$ 、 $V_{t,\max}$ 分别对应时段 t 水库库容的最小值和最大值； r 对应随机数 (0~1)。

以水库库容进行实数编码，基因段变化范围为水库水位变化范围对应的库容，随机选取 pop 组可行库容序列 $(V_1^1, V_2^1, \dots, V_n^1)$, $(V_1^2, V_2^2, \dots, V_n^2)$, ..., $(V_1^{pop}, V_2^{pop}, \dots, V_n^{pop})$ ，其中， pop 为种群规模， n 为对应的时段数。

水库调度目标函数是多峰值函数，以算法迭代次数为横坐标，目标函数值为纵坐标，其完整的曲线应该是一个多峰值曲线。为了体现分层优化算法优越性，与自适应遗传算法以及广度变异算法进行比较，通过多次实验结果，迭代次数设置为 100 代是合理的，优化结果如图 3（图中的改进算法优化结果从第二层开始迭代）。

从图 3 可以看出：①自适应遗传算法存在“早熟”的缺点，在迭代过程中极早陷入了局部最优，而且不具备跳出局部最优能力，得到最优值并不是调度问题中真正最优值，可能是对应的精英个体，因此得到最

优结果不够理想，而单纯采用随机种群，大大降低了种群的多样性，大大限制了遗传算法收敛到全局最优。②在自适应遗传算法中嵌入“广度变异”模块，通过增加种群的多样性在一定程度上帮助算法跳出局部最优，指导遗传算法进入新的搜索空间，有效地改善了最优解的质量。③通过分层自适应遗传算法计算结果可知，由于第一层为了增加种群的多样性，保留了部分不可行解，而且由于分层算法中广度变异模块，导致分层遗传算法不停地进化收敛，这是一个不断寻优过程。分层的思想是通过广度变异和外部存档增加了种群的多样性，从而增加了算法的鲁棒性，抑制了遗传算法“早熟”现象，从而改善了进化结果。

3.3 两种算法的结果比较

分别采用并行动态规划算法和分层遗传算法求解水库优化调度模型。

1) 水库水位变化

由图 4 可知，动态规划算法的放水过程并不均匀，枯水年来水较少，为了满足保证出力要求水库水位降得较快；而分层算法的放水比较均匀，水位变化平缓，维持高水位从而得到较好发电效果。

2) 水库出力

从图 5 可以看出，12 月份以前两种算法对应水库出力基本相同，都满足保证出力，而 12 月份到翌年 4 月份，分层算法基本以保证出力进行运行，而分层算法的调度过程中出力则小于保证出力。

3) 从可靠性角度讲，由于选用枯水年的资料，这两种算法优化结果都很难满足下泄流量的要求。而对应单个时段，动态规划优化调度中最小出力基本上满足保证出力，而分层算法并没有满足可靠性要求，降低水库运行的可靠性。

4) 从经济性角度来讲，整个非汛期的发电量分层算法的 496.34 亿 kW·h 明显要高于动态规划的 487.7 亿 kW·h，分层算法调度效益较好。

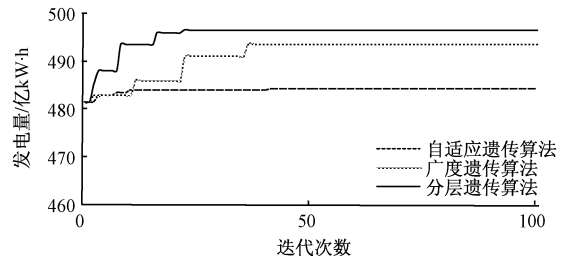


图 3 三种算法优化结果
Fig.3 Optimized results with the three algorithms

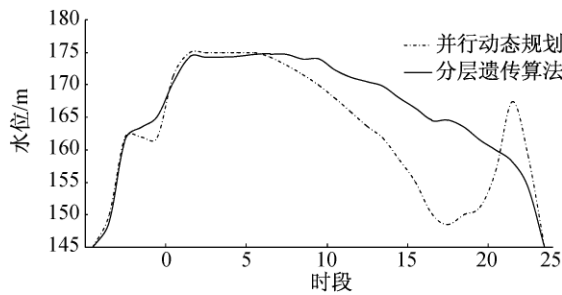


图 4 优化调度的水库水位
Fig.4 Reservoir water levels of optimal operation

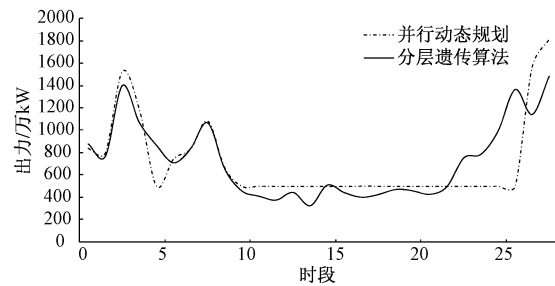


图 5 优化调度的水库出力
Fig.5 Power outputs of optimal operation

3.4 长系列计算结果比较

为了验证改进算法的有效性，选取三峡库区 11 年入库径流作为模型输入，其来水频率分别为 85%~95%，并采用并行计算线程为 4，并行算法的计算时间比较以及各算法多年平均发电量比较如表 2、3。

表 2 串并行计算时间比较
Table 2 Comparison of elapsed clock time by serial and parallel computations

计算次数	1	2	3	平均值
串行计算	350.95	348.67	348.73	349.45
并行计算	93.61	93.74	93.68	93.67

从表 2 可以看出，并行动态规划算法充分利用计算机资源，使工作并行进行，从而缩短计算时间，大

大提高了计算效率。特别对于长系列模拟，庞大的计算工作量，更能体现并行计算的优越性。

从表3可以看出，分层算法得到的多年平均发电量明显优于其它两种算法，说明分层算法提高了调度效益。

表3 优化算法的多年平均发电量结果比较
Table 3 Comparison of average annual power yields calculated by optimal algorithms 亿 kW·h

算法	自适应算法	广度变异算法	分层算法
发电量	559.98	561.32	565.72

4 结论

基于动态规划算法，嵌入并行计算模块；在自适应遗传算法基础上，采用外部存档的方式，提出了改进的分层遗传算法，并将改进算法应用到三峡水库实际调度中。实验算例的应用情况表明，并行动态规划算法相比于串行计算，提高了算法计算效率；与其它两个算法相比较，分层遗传算法提高了收敛速度并改善了调度效益，而长系列的计算结果进一步验证了改进算法的有效性。改进的算法从不同的角度提高了求解速度并改善了求解质量，为水库优化调度问题的求解提供了一定的参考。

参考文献：

- [1] Bellman R.R. Dynamic Programming [M]. New Jersey, Princeton University Press, 1957:3-59.
- [2] 涂启玉, 梅亚东. 遗传算法在水库(群)优化调度研究中的应用综述 [J]. 水电自动化与大坝监测, 2008, 32(1): 16-18.
TU Qiyu, MEI Yadong. Review on the application of genetic algorithm to reservoirs optimal operation [J]. Hydropower Automation & Dam Monitoring, 2008, 32(1): 16-18. (in Chinese)
- [3] 毛睿, 黄刘生, 徐大杰, 等. 淮河中上游库群联合优化调度算法及并行实现 [J]. 小型微型计算机系统 2000, 21(6): 603-607.
MAO Rui, HUANG Liusheng, XU Dajie, et al. Algorithm and implementation of multi-reservoir optimal [J]. Journal of Mini Micro System, 2000, 21(6): 603-607. (in Chinese)
- [4] 马光文, 王黎. 遗传算法在水电站优化调度中应用 [J]. 水科学进展, 1997, 8(3): 275-280.
MA Guangwen WANG Li. Application of a genetic algorithm to optimal operation of hydropower station [J]. Advances in Water Science, 1997, 8(3): 275-280. (in Chinese)
- [5] 畅建霞, 黄强, 王义民. 基于改进遗传算法的水电站水库优化调度[J]. 水力发电学报, 2001, 3: 85-90.
CHANG Jianxia, HUANG Qiang, WANG Yimin. Optimal operation of hydropower station reservoir by using an improved Genetic Algorithm [J]. Journal of Hydroelectric Engineering, 2001, 3: 85-90. (in Chinese)
- [6] 陈立华, 梅亚东, 董雅洁, 等. 改进遗传算法及其在水库群优化调度中的应用 [J]. 水利学报, 2008, 39(5): 550-556.
CHEN Lihua, MEI Yadong, DONG Yajie, et al. Improved genetic algorithm and its application in optimal dispatch of cascade reservoir[J]. Journal of Hydraulic Engineering, 2008, 39(5): 550-556. (in Chinese)
- [7] 王少波, 解建仓, 孔珂. 自适应遗传算法在水库优化调度中的应用 [J]. 水利学报, 2006, 37(4): 480-485.
WANG Shaobo, XIE Jiancang, KONG Ke. Application of adaptive genetic algorithm in optimization of reservoir operation [J]. Journal of Hydraulic Engineering, 2006, 37(4): 480-485. (in Chinese)
- [8] Akhter. S, Roberts. J. Multi-core programming: increasing performance through software multithreading [M]. Hillsboro, Oregon: Intel Press, 2006:1-19.
- [9] 程春田, 郜晓亚, 武新宇, 等. 梯级水电站长期优化调度的细粒度并行离散微分动态规划方法[J]. 中国电机工程学报, 2011, 31(10): 26-32.
CHENG Chuntian, GAO Xiaoya, WU Xinyu, et al. Fine-grained parallel discrete differentiation and dynamic programming algorithm for long-term optimization of cascade hydropower system [C]//Proceeding of the CSEE, 2011, 31(10): 26-32. (in Chinese)
- [10] 陈立华, 梅亚东, 麻荣永. 并行遗传算法在雅砻江梯级水库群优化调度中应用[J]. 水力发电学报, 2010, 29(6): 66-70.
CHEN Lihua, MEI Yadong, MA Rongyong. Parallel genetic algorithm and its application to optimal operation of the Yalong river cascade reservoirs [J]. Journal of Hydroelectric Engineering, 2010, 29(6): 66-70. (in Chinese)