algorithms

MDPI

Article

A Gradient-Based Cuckoo Search Algorithm for a Reservoir-Generation Scheduling Problem

Yu Feng [1,2], Jianzhong Zhou [1,2,*], Li Mo [1,2], Chao Wang [3], Zhe Yuan [4] and Jiang Wu [4]

1 School of Hydropower and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China; ferrychip@hust.edu.cn (Y.F.); moli@hust.edu.cn (L.M.)
2 Hubei Key Laboratory of Digital Valley Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
3 China Institute of Water Resources and Hydropower Research, Beijing 100038, China; c.wang1222@gmail.com
4 Changjiang River Scientific Research Institute, Changjiang Water Resources Commission of the Ministry of Water Resources of China, Wuhan 430010, China; yuanzhe_0116@126.com (Z.Y.); cky_wujiang@163.com (J.W.)
* Correspondence: jz.zhou@hust.edu.cn; Tel.: +86-027-8754-3127

Received: 22 January 2018; Accepted: 20 March 2018; Published: 25 March 2018

Abstract: In this paper, a gradient-based cuckoo search algorithm (GCS) is proposed to solve a reservoir-scheduling problem. The classical cuckoo search (CS) is first improved by a self-adaptive solution-generation technique, together with a differential strategy for Lévy flight. This improved CS is then employed to solve the reservoir-scheduling problem, and a two-way solution-correction strategy is introduced to handle variants' constraints. Moreover, a gradient-based search strategy is developed to improve the search speed and accuracy. Finally, the proposed GCS is used to obtain optimal schemes for cascade reservoirs in the Jinsha River, China. Results show that the mean and standard deviation of power generation obtained by GCS are much better than other methods. The converging speed of GCS is also faster. In the optimal results, the fluctuation of the water level obtained by GCS is small, indicating the proposed GCS's effectiveness in dealing with reservoir-scheduling problems.

Keywords: long-term hydropower generation scheduling; cascade reservoirs; gradient-based cuckoo search algorithm; Jinsha River

1. Introduction

The penetration of renewable power generation has increased substantially in recent years [1]. The use of renewable energy such as wind power and hydropower can reduce greenhouse gas emissions from fossil fuels. Wind power is one of the fastest-growing sources, however it is uncertain and cannot be scheduled due to its intrinsic dependence on varying weather conditions over many years [2,3]. Solar power is less uncertain, but has higher variability compared to wind power [4]. Unlike wind energy, hydropower can be scheduled and has already played an important role in electricity supply all over the world. The long-term hydropower-generation scheduling (LHGS) of cascade reservoirs is of great importance to improving power grid stability. A suitable scheduling scheme can effectively increase the economic benefit of cascade reservoirs. The purpose of LHGS is to determine the water release of all reservoirs in a time period in order to maximize the benefit [5].

Many studies have been carried out [6–8] indicating that the LHGS is a non-linear and non-convex problem with very complicate constraints including water balancing, hydraulic connection, water-level limitations, water-release limitations and output capacity limitations [9]. In order to solve this problem, classical studies have taken many kinds of approaches, including dynamic programming

(DP) [10,11], linear programming (LP) [12,13], the progressive optimality algorithm (POA) [14,15], and discrete differential dynamic programming (DDDP) [16,17]. However, these methods have their own shortcomings in dealing with the problems of large-scale cascade reservoirs [18].

In recent years, meta-heuristic algorithms have been popular in the search for a global optimum in non-convex problems [19]. Metaheuristics and their applications have wide application in artificial intelligence, e.g., water-distribution systems [20], transit network design problems [21], reservoir-operation optimization [22], and so forth. Many metaheuristics have included some form of stochastic optimization, so the searching efficiency is affected by the set of random variables generated [23]. By searching over a large numbers of feasible solutions, metaheuristics can often find good solutions near to the optimum solution with less computational effort [24].

Cuckoo search (CS) is one kind of metaheuristics developed by Xin-she Yang and Suash Deb in 2009 [25]. It was inspired by the behavior of some cuckoo species by laying eggs in the nests of other host birds. Compared to other artificial intelligence algorithms such as the genetic algorithm (GA) [26] and particle swarm optimization (PSO) [27], CS has the characteristics of higher solution quality, shorter computational time, and higher success rates [28]. It has been applied to various optimization problems. For example, Maiya Din et al. introduced it for the analysis of linear feedback shift register-based cryptosystems [29], Bhateja et al. used CS for the cryptanalysis of vigenere [30], Naik et al. used CS for intrinsic discriminant analysis-based face recognition [31].

Although CS has shown its advantages and has wide application in many areas, it still has some drawbacks. Proper computational time cannot be guaranteed in large-scale systems [32]. In this paper, some improvements are proposed to the original CS using four strategies: (1) a dynamic parameter adjustment of the parameter; (2) a boundary value perturbation strategy for boundary value adjustment; (3) a differential strategy for Lévy flight; (4) a different way to update solutions. When applying CS to the LHGS problem, we find that the converge speed and the results are not satisfactory. What's more, solutions generated by a random search have the characteristic of large fluctuations in water level. Thus, a gradient strategy is designed especially for the LHGS problem. This gradient strategy improves the local search ability of the algorithm by fine-tuning the water level process. In case studies, comparisons of benchmark function tests among different algorithms has been carried out. Results show that the improved cuckoo search (ICS) performs much better than the improved harmony search (IHS) and CS. Finally, a case study of the Jinsha River is put forward to show the application of the LHGS model. The advantage of the proposed GCS is very obvious: it can effectively obtain satisfactory results in all wet years, normal years and dry years.

This paper is organized as follows: Section 2 describes the objective function and constraints of LHGS model; Section 3 gives the details of improved CS; Section 4 presents the gradient-based CS algorithm for LHGS; Section 5 is the experiments and the case study in the Jinsha River; and Section 6 is the summary of this paper.

2. Reservoir-Scheduling Problem

The main purpose of LHGS is to improve the power generation of cascaded reservoirs. The objective function can be listed as follows.

2.1. Long-Term Hydropower-Generation Scheduling (LHGS) Model

The LHGS model is used to find a set of water releases or storage volumes that maximizes the generating benefit for cascade hydropower system. The objective formula of hydropower optimization problem is given as follows:

$$E = \max \sum_{t=1}^{T} \sum_{i=1}^{N} k_i Q_{i,t} H_{i,t} \Delta t \quad (1)$$

where E is the total energy production of the cascade hydropower system; T is the count of periods; N is the number of reservoirs; k_i is the output coefficient of reservoir i; $Q_{i,t}$ is the generation flow

which outflows through hydropower units of the reservoir i in time t; $H_{i,t}$ is the net water head of hydropower reservoir i in time t; and Δt is the length of time interval.

2.2. Constraints

To complete the scheduling model, the constraints of the scheduling problem are defined as:

(1) Hydraulic Connection

$$\begin{cases} I_{i,t} = O_{i-1,t} + R_{i,t} \\ O_{i-1,t} = Q_{i-1,t} + S_{i-1,t} \end{cases} \tag{2}$$

$I_{i,t}$ is the inflow of reservoir i in time t; $O_{i-1,t}$ is the outflow of reservoir $i-1$ in time t; $R_{i,t}$ is the interval inflow between reservoir $i-1$ and i; $S_{i-1,t}$ is spillage of the upstream reservoir $i-1$; $Q_{i-1,t}$ is the generation flow of the upstream reservoir $i-1$.

(2) Water-Balance Constraint

$$V_{i,t} = V_{i,t-1} + (I_{i,t} - O_{i,t}) \cdot \Delta t \tag{3}$$

$V_{i,t}$ is the reservoir storage of reservoir i in time t.

(3) Water-Level Constraints

$$Z_{i,t}^{\min} \le Z_{i,t} \le Z_{i,t}^{\max} \tag{4}$$

$$|Z_{i,t} - Z_{i,t-1}| \le Z_{i,t}^{step} \tag{5}$$

$Z_{i,t}^{min}$ and $Z_{i,t}^{max}$ are, respectively, the lower and upper bounds of the water level; Z_i^{step} is the limitation of water-level variation.

(4) Outflow Constraint

$$O_{i,t}^{\min} \le O_{i,t} \le O_{i,t}^{\max} \tag{6}$$

$O_{i,t}^{min}$ and $O_{i,t}^{max}$ are, respectively, the minimum and maximum outflows of reservoir i.

(5) Output Constraint

$$N_{i,t}^{\min} \le N_{i,t} \le N_{i,t}^{\max} \tag{7}$$

$N_{i,t}^{min}$ and $N_{i,t}^{max}$ are, respectively, the minimum and maximum power output of reservoir i.

(6) Boundary Condition

$$Z_{i,0} = Z_{i,start}, Z_{i,T} = Z_{i,end} \tag{8}$$

$Z_{i,start}$ and $Z_{i,end}$ are, respectively, the initial and terminal water levels of reservoir i.

3. Cuckoo Search Algorithm

3.1. Basic Cuckoo Search Algorithm

Cuckoo search (CS) is a meta-heuristic algorithm inspired by the obligate brood parasitism of some cuckoo species [25]. It is a population-based evolutionary algorithm and can be applied for various optimization problems. The procedures of CS are explained as follows:

1. Each cuckoo lays one egg at a time, and dumps its egg in randomly chosen nest;
2. The best nests with a high quality of eggs will carry over to the next generation;
3. The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability p_a. Discovering operates on some set of the worst nests, and discovered solutions are dumped from further calculations.

For a minimum function, the basic steps of the CS are summarized as the pseudo code in Algorithm 1:

Algorithm 1. Cuckoo search via Lévy flights

Objective function $f(\mathbf{x}), \mathbf{x} = (x_1, ..., x_d)^T$
Generate initial population of n host nests $\mathbf{x}_i (i = 1, 2, ..., n)$
While ($t <$ *MaxIteration*) or (*stop criterion*)
  Get a cuckoo i randomly by Lévy flights and evaluate its fitness
  Choose a nest j among n randomly
  If $f(\mathbf{x}_i) < f\left(\mathbf{x}_j\right)$
    Replace $\mathbf{x}_j$ by the new solution $\mathbf{x}_i$
  End if
  If $rand(0, 1) < p_a$
    Init the worst nest $\mathbf{x}_{worst}$
  End if
  If $f(\mathbf{x}_i) < f(\mathbf{x}_{best})$
    Replace $\mathbf{x}_{best}$ by $\mathbf{x}_i$
  End if
End while

3.2. Improvement for Cuckoo Search Algorithm

The basic cuckoo search algorithm (CS) is capable of finding the optimum function, but after several iterations, we found that it still needs improvements. In this section, three strategies are presented to improve the CS.

3.2.1. Dynamic Parameter Adjustment Strategy

In the CS algorithm, the abandonment of the nest is determined by its abandon probability p_a. If p_a is large, this nest is considered as a bad solution and would be replaced with new randomly generated solutions. The converging speed of the algorithm will decline due to the replacement, but the population diversity can be improved. If p_a is small, the characteristics of the population can be reserved and it can benefit the convergence. According to this feature, a dynamic adjustment strategy is introduced to decide p_a showing as Equation (9). With this strategy, there would be better global search capabilities in the early stage evolution, and the convergence becomes faster in the later stage:

$$p_a = p_a^s + (p_a^e - p_a^s) \times \frac{CE}{NE} \tag{9}$$

where CE is the current evaluation times; NE is the maximum evaluation times; and p_a^s and p_a^e are start and end p_a, respectively.

3.2.2. A Boundary Value Perturbation Strategy

When a new solution is generated, the values of some dimensions may be no longer in the proper range. The common method is to adjust it to the boundary value, which may result in unexpected accumulation of solutions on the border. Thus, a boundary value perturbation strategy is carried out, as shown in Equation (10):

$$x_i^d(t) = \begin{cases} UB^d - rand(0,1) \times \mathrm{mod}\left(x_i^d(t) - UB^d, UB^d - LB^d\right), \text{ if } x_i^d(t) > UB^d \\ LB^d + rand(0,1) \times \mathrm{mod}\left(LB^d - x_i^d(t), UB^d - LB^d\right), \text{ if } x_i^d(t) < LB^d \end{cases} \tag{10}$$

where mod$(a,\ b)$ is modulo operation to find the remainder after division of a by b; and UB^d and LB^d are the upper and lower boundary limits, respectively.

3.2.3. Differential Strategy for Lévy Flight

In CS, an important problem is the implementation of Lévy flights, as shown in the original paper [25]:

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \alpha \oplus Levy(\lambda) \tag{11}$$

Unfortunately, the original paper only described the basic idea of Lévy flights, and the details of its implementation were not given. Later studies had different interpretations of Lévy flight such as [33,34]. In this paper, the strategy used for Lévy flight is given as:

$$x_i^{d\prime} = x_i^d + \left(sl \times Levy(u,c) \times \left(x_j^d - x_i^d\right)\right) \tag{12}$$

where i and j are the numbers of randomly chosen nests; sl is the step size parameter; $Levy(u,c)$ is sample value from the Lévy distribution, and its probability density function is:

$$f(x) = \sqrt{\frac{c}{2\pi}}\frac{e^{-\frac{c}{2(x-u)}}}{(x-u)^{3/2}} \tag{13}$$

3.2.4. Solution Updates Strategy Changes

The CS generates new solutions by Lévy flights and compares it with another randomly chosen nest. However, when replacing the randomly chosen solution, the source solution and the target solution are not relevant, so the useful information from the target solution is missing. Thus, if the new generated solution is better, instead of replacing another randomly chosen nest, we replace the source nest that is used for performing Lévy flights.

Additionally, in the proposed algorithm, the guide information from the best solution or current best solution is not used, and we should not find the optimal solution in each iteration.

3.3. Implementation of Improved Cuckoo Search (ICS)

With the improvements above, the improved CS can be described as follows:

Step 1 Initialize the algorithm parameters

This step specifies the desired parameters, including size of population (ns), abandon probability (p_a), disturbance weight (w), step size (sl), and Lévy distribution parameters u and c.

Step 2 Initialize the population

In this step, new random nests ($\mathbf{x}_1, ..., \mathbf{x}_{ns}$), whose total number is ns, are generated as in Equation (14):

$$x_i^d = (UB^d - LB^d) \cdot rand(0,1) + LB^d, d = 1, 2, ..., D \tag{14}$$

Then each vector will be evaluated by the objective function and stored in the population.

Step 3 Generate a new solution by Lévy flights

Here a source nest $\mathbf{x}_i$ and target nest $\mathbf{x}_j$ from the population are randomly chosen. A new cuckoo from nest i flight to nest j obeys the features of Lévy flight. Then a new solution is generated from Equation (12).

If the new generated solution is out of proper range, we would adjust the solution from Equation (10).

Step 4 Update the host nest

Evaluate the fitness of the new solution $f(\mathbf{x}'_i)$ and compare it with the fitness of source nest $f(\mathbf{x}_i)$. If the new solution is better, replace the source nest with the new solution.

Step 5 Abandon the worst nest

If the random value is smaller than p_a, a new solution is generated randomly and it will replace the worst nest.

Step 6 Check the stop criterion

Repeat Step 3 to Step 5 until the termination criterion is satisfied. In this paper, the termination criterion is the maximum number of evaluation times.

The pseudo-code of the ICS algorithm is presented in Algorithm 2.

Algorithm 2. Improved cuckoo search

Objective function $f(\mathbf{x}), \mathbf{x} = (x_1, ..., x_d)^T$
Initialize default parameters
Generate initial population of n host nests $\mathbf{x}_i (i = 1, 2, ..., n)$
While (t < *MaxEvaluation*) or (*stop criterion*)
  Select two solution $\mathbf{x}_i, \mathbf{x}_j$ from host nests randomly
  For d=1,...,D **do**
    $x_i^{d\prime} = x_i^d + \left(sl \times Levy(u, c) \times \left(x_j^d - x_i^d\right)\right)$
  End for
  If $f(\mathbf{x}'_i) < f(\mathbf{x}_i)$
    Replace $\mathbf{x}_i$ by the new solution $\mathbf{x}'_i$
  End if
  If $rand(0, 1) < p_a$
    Init the worst nest $\mathbf{x}_{worst}$
  End if
End while

4. Gradient Cuckoo Search for Reservoir Scheduling

Generally, when the heuristic algorithm is applied on the reservoir-scheduling problem, the process of water levels is selected as the decision variable, as shown in Equation (15):

$$X = \begin{bmatrix} X_1 \\ X_2 \\ \cdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_1^1, \cdots x_1^d, \cdots x_1^T \\ x_2^1, \cdots x_2^d, \cdots x_2^T \\ \cdots \\ x_N^1, \cdots x_N^d, \cdots x_N^T \end{bmatrix} \tag{15}$$

where N is the number of reservoirs; and T is the number of time intervals.

4.1. Constraints Handling

Cascade reservoir operation is a complicated optimization problem with varieties of constraints. According to ICS, the initial population is generated following Equation (16):

$$x_i = x_i^{\min} + rand \cdot \left(x_i^{\max} - x_i^{\min}\right) \tag{16}$$

where x_i^{min} and x_i^{max} are, respectively, the lower and upper boundaries limit of reservoir i; and *rand* is a random number between 0 and 1. However, there is a high probability that the randomly generated solutions do not satisfy all the constraints displayed in Section 2.2. In the initialization stage, if the generated solution is not feasible, it needs to be replaced by a new feasible solution.

In the random search stage, the newly generated unfeasible solution cannot be discarded. It is adjusted to the feasible range as follows:

$$x_{i+1} = \begin{cases} x_{i+1}^{\min}, if\ x_{i+1} < x_{i+1}^{\min} \\ x_{i+1}^{\max}, if\ x_{i+1} > x_{i+1}^{\max} \end{cases} \tag{17}$$

The feasible range of water level x_{i+1} is defined as:

$$x_{i+1}^{\max} = \min\left(x_i + \Delta z, Z\left(x_i, Q_i^{\min}\right), Z\left(x_i, Q_i^{P\min}\right), z^{\max}\right) \tag{18}$$

$$x_{i+1}^{\min} = \max\left(x_i - \Delta z, Z(x_i, Q_i^{\max}), z^{\min}\right) \tag{19}$$

$$Z(x_i, Q_i) = Z(V(x_i) + (I_i - Q_i)\Delta t) \tag{20}$$

where Q_t^{min} and Q_t^{max} are, respectively, the minimum and maximum outflows of the reservoir; Q_t^{Pmin} is the minimum outflow for a guaranteed output; z^{min} and z^{max} are minimum and maximum water levels; Δz is the limitation of the water-level change; and $Z(V)$ and $V(Z)$ are the relationship between the water level and the storage.

In the adjustment of the unfeasible solutions, a two-way solution correction strategy is employed. First, the water level is adjusted to the feasible range from period 1 to T. If the correction fails at some unpredictable period, then the water level is adjusted from period T to the breakpoint. If the correction is still not successful, the solution would be considered as an infeasible solution.

4.2. Gradient-Based Search Strategy

In the random search phase, new solutions are randomly generated from the heuristics algorithms. However, after many numerical experiments, it is found that the performances of the heuristic algorithms fail to meet the requirements. On the one hand, the variance of the calculation results is large, indicating that the algorithms are not very stable. On the other hand, the converging speed of the algorithm is quite slow when dealing with the complex cascade reservoirs problem. Hence in this section, a gradient-based search strategy is designed especially for LHGS.

For a random generated solution, the water level is adjusted by a small gradient $\Delta l_{i,t}$ as shown in Figure 1, which would result in the power generation of the entire cascade reservoir changing.

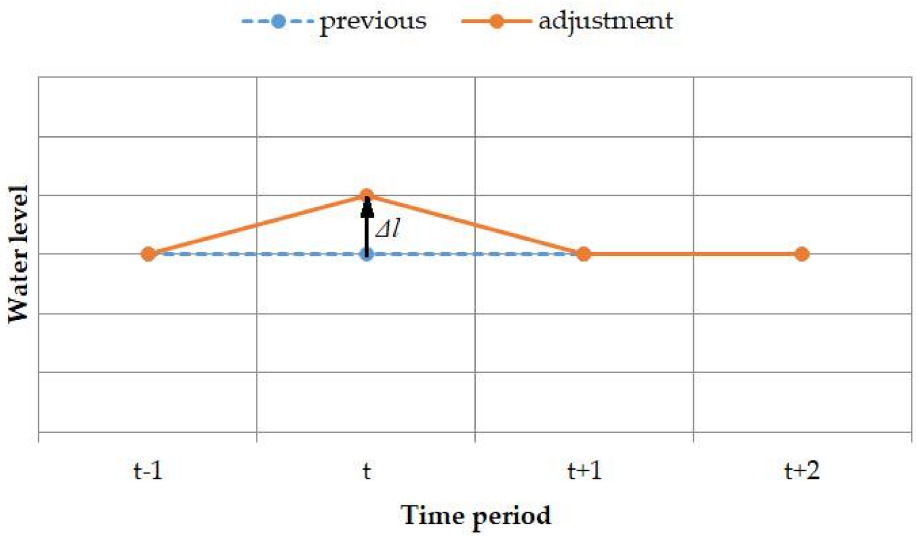


Figure 1. Adjustment in water level by a small gradient.

If we change the end water level of reservoir i by Δl at time period t, the power generation of reservoir i at time period t is:

$$\begin{cases} Z_{i,t}^{adjust} = Z_{i,t} + \Delta l \\ \Delta Q_{i,t} = (V_i(Z_{i,t} + \Delta l) - V_i(Z_{i,t}))/\Delta T_t \\ \Delta H_{i,t} = \Delta l/2 - Zd_i(Q_{i,t} + \Delta Q_{i,t}) + Zd_i(Q_{i,t}) \\ \Delta P_{i,t} = k_i(Q_{i,t} + \Delta Q_{i,t})(H_{i,t} + \Delta H_{i,t}) - k_i Q_{i,t} H_{i,t} \end{cases} \tag{21}$$

where $V_i(Z)$ is the fitting function of water level and storage curve of reservoir i; and $Zd_i(Q)$ is the fitting function of the downstream water level and outflow curve of reservoir i.

Then the power generation of reservoir i at the next time period $t + 1$ can be given by:

$$\begin{cases} \Delta Q_{i,t+1} = (V_i(Z_{i,t}) - V_i(Z_{i,t} + \Delta l))/\Delta T_{t+1} \\ \Delta H_{i,t+1} = \Delta l/2 - Zd(Q_{i,t+1} + \Delta Q_{i,t+1}) + Zd(Q_{i,t+1}) \\ \Delta P_{i,t+1} = k_i(Q_{i,t+1} + \Delta Q_{i,t+1})(H_{i,t+1} + \Delta H_{i,t+1}) - k_i Q_{i,t+1} H_{i,t+1} \end{cases} \tag{22}$$

When the outflow of reservoir i is changed and the water level of the downstream reservoirs remains the same, the inflow of all reservoirs in the downstream reservoir would be changed by the outflow of reservoir i. For one of the reservoirs downstream, the power generation at time period t is:

$$\begin{cases} \Delta Q_{j,t} = \Delta Q_{i,t} \\ \Delta H_{j,t} = Zd_j(Q_{j,t}) - Zd_j(Q_{j,t} + \Delta Q_{j,t}) \\ \Delta P_{j,t} = k_j(Q_{j,t} + \Delta Q_{j,t})(H_{j,t} + \Delta H_{j,t}) - k_j Q_{j,t} H_{j,t} \end{cases} \tag{23}$$

The power generation of the next reservoir $i + 1$ at time period $t + 1$ is:

$$\begin{cases} \Delta Q_{j,t+1} = \Delta Q_{i,t+1} \\ \Delta H_{j,t+1} = Zd_j(Q_{j,t+1}) - Zd_j(Q_{j,t+1} + \Delta Q_{j,t+1}) \\ \Delta P_{j,t+1} = k_j(Q_{j,t+1} + \Delta Q_{j,t+1})(H_{j,t+1} + \Delta H_{j,t+1}) - k_j Q_{j,t+1} H_{j,t+1} \end{cases} \tag{24}$$

For the solution adjustment, we need to adjust this to the better direction. Instead of a time-consuming fitness calculation of the incremental power generation ΔE, we only need to determine whether the incremental power generation ΔE is positive. If $\Delta E > 0$, the adjustment will be made to give a better solution. Otherwise the previous solution is kept without change.

In this paper, we determine whether ΔE is positive from the calculation of the partial derivative of power generation E with respect to the gradient Δl, as given below. For the reservoir i at time period t shown in Equation (25), the partial derivative of power output is:

$$\begin{cases} \frac{\partial Q_{i,i}}{\partial l} = \lim\limits_{\Delta l \to 0} \frac{\Delta Q_{i,t}}{\Delta l} = \frac{1}{\Delta T_t} \lim\limits_{\Delta l \to 0} \frac{V_i(Z_{i,t} + \Delta l) - V_i(Z_{i,t})}{\Delta l} = \frac{1}{\Delta T_i} \cdot V'_i(Z_{i,t}) \\ \frac{\partial H_{i,t}}{\partial l} = \lim\limits_{\Delta l \to 0} \frac{\Delta l/2 - Zd_i(Q_{i,t} + \Delta Q_{i,t}) + Zd_i(Q_{i,t})}{\Delta l} = \frac{1}{2} - Zd'_i(Q_{i,t}) \cdot \frac{\partial Q_{i,t}}{\partial l} \\ \frac{\partial P_{i,t}}{\partial l} = k_i \cdot \left(Q_{i,t} \frac{\partial H_{i,t}}{\partial l} + H_{i,t} \frac{\partial Q_{i,t}}{\partial l} \right) \end{cases} \tag{25}$$

For the reservoir i at time period $t + 1$ shown in Equation (26), the partial derivative of power output is:

$$\begin{cases} \frac{\partial Q_{t+1}}{\partial l} = \lim\limits_{\Delta l \to 0} \frac{V(Z_t) - V(Z_t + \Delta l)}{\Delta l \Delta T_{t+1}} = -\frac{1}{\Delta T_{t+1}} \cdot V'(Z_t) \\ \frac{\partial H_{t+1}}{\partial l} = \frac{1}{2} - \lim\limits_{\Delta l \to 0} \frac{Zd(Q_{t+1} + \Delta Q_{t+1}) - Zd(Q_{t+1})}{\Delta l} = \frac{1}{2} - Zd'(Q_{t+1}) \cdot \frac{\partial Q_{t+1}}{\partial l} \\ \frac{\partial P_{t+1}}{\partial l} = k \cdot \left(Q_t \frac{\partial H_{t+1}}{\partial l} + H_t \frac{\partial Q_{t+1}}{\partial l} \right) \end{cases} \tag{26}$$

For the reservoir downstream at time period t shown in Equation (27), the partial derivative of power output is:

$$\begin{cases} \frac{\partial Q_{j,t}}{\partial l} = \frac{\partial Q_{i,t}}{\partial l} \\ \frac{\partial H_{j,t}}{\partial l} = \lim\limits_{\Delta l \to 0} \frac{Zd_j(Q_{j,t}) - Zd_j(Q_{j,t} + \Delta Q_{j,t})}{\Delta l} = -Zd'_j(Q_{j,t}) \cdot \frac{\partial Q_{j,t}}{\partial l} \\ \frac{\partial P_{j,t}}{\partial l} = k_j \cdot \left(Q_{j,t} \frac{\partial H_{j,t}}{\partial l} + H_{j,t} \frac{\partial Q_{j,t}}{\partial l} \right) \end{cases} \tag{27}$$

For the reservoir downstream at time period $t + 1$ shown in Equation (28), the partial derivative of power output is:

$$\begin{cases} \frac{\partial Q_{j,t+1}}{\partial l} = \frac{\partial Q_{i,t+1}}{\partial l} \\ \frac{\partial H_{j,t+1}}{\partial l} = -Zd'_j(Q_{j,t+1}) \cdot \frac{\partial Q_{j,t+1}}{\partial l} \\ \frac{\partial P_{j,t+1}}{\partial l} = k_j \cdot \left(Q_{j,t+1} \frac{\partial H_{j,t+1}}{\partial l} + H_{j,t+1} \frac{\partial Q_{j,t+1}}{\partial l} \right) \end{cases} \tag{28}$$

According to the formulas above, the partial derivative of power generation E with respect to the gradient Δl is:

$$\frac{\partial E}{\partial l} = \Delta T_t \frac{\partial P_{i,t}}{\partial l} + \Delta T_{t+1} \frac{\partial P_{i,t+1}}{\partial l} + \sum_{j=i+1}^{N} \left(\Delta T_t \frac{\partial P_{j,t}}{\partial l} + \Delta T_{t+1} \frac{\partial P_{j,t+1}}{\partial l} \right) \tag{29}$$

If $\frac{\partial E}{\partial l} > 0$, adjust the water level by Δl, otherwise adjustment is rejected. For the cascade system, the flowchart of the gradient-based search strategy is shown in Figure 2.

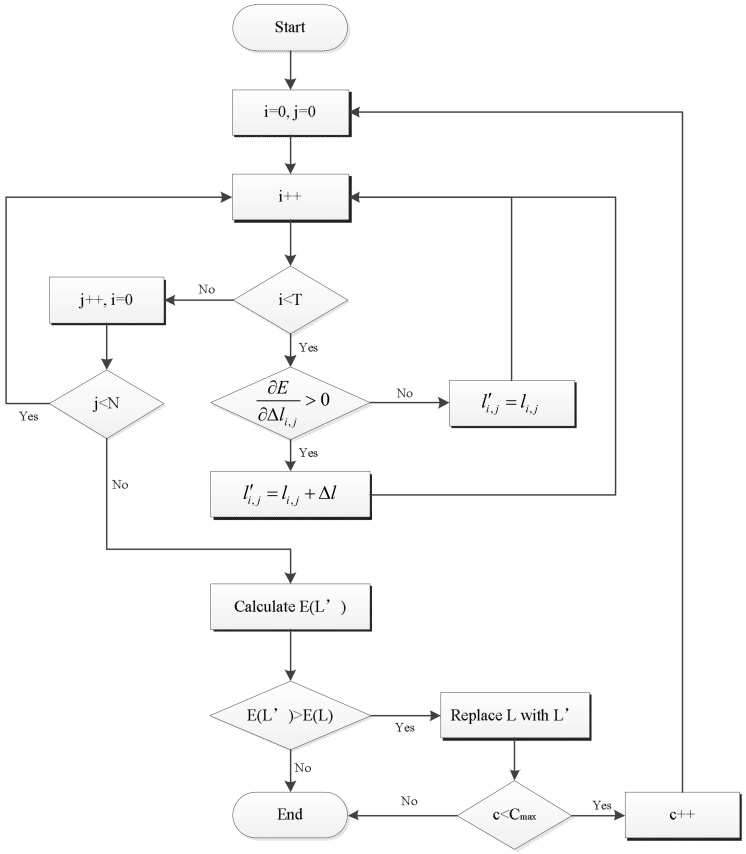


Figure 2. Flowchart of gradient-based search strategy.

4.3. Implementation of Gradient-Based Cuckoo Search (GCS) for Long-Term Hydropower Generation Scheduling (LHGS)

According to the description above, the steps to solve long-term scheduling problem using the cuckoo search algorithm with the gradient-based search strategy (GCS) are shown in Figure 3:

Step 1: Randomly generate feasible initial solutions.
Step 2: Evaluate fitness of the solutions.
Step 3: Generate new solutions by Lévy flights.
Step 4: If the new solution is infeasible, adjust it by two-way solution correction strategy.
Step 5: Adjust the new solution by gradient-based search strategy.
Step 6: Update the host nest.
Step 7: Abandon the worst nest.
Step 8: Repeat Steps 3 to 7 until the stop criteria is reached.

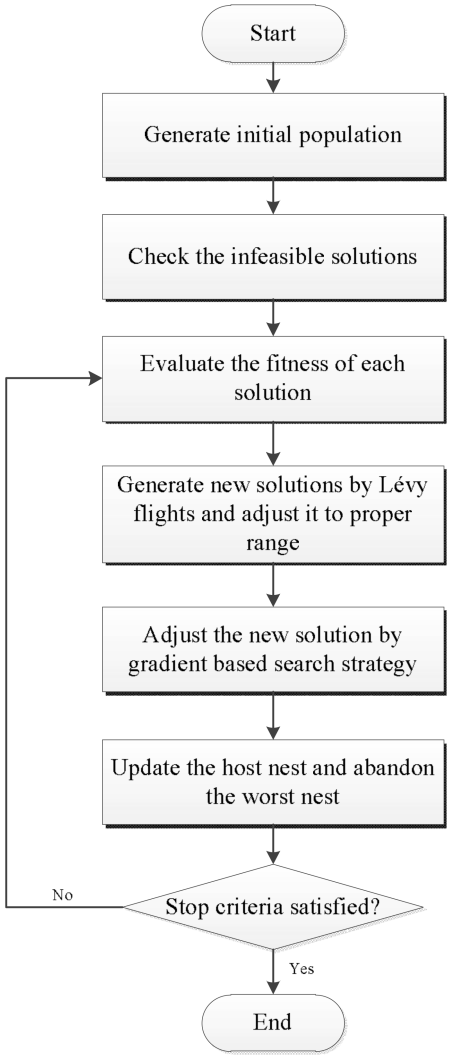


Figure 3. Flowchart of gradient-based cuckoo search (GCS).

5. Case Study

5.1. Study Area

The Jinsha River is the upper reach of the Yangtze River in south-west China. It flows through Qinghai, Sichuan and Yunnan provinces. The Jinsha River is rich in water resources, and is being heavily developed for hydroelectric power. In this case study, we discuss four hydropower stations with total installed capacity of 42,960 MW on the river, including Wudongde, Baihetan, Xiluodu and Xiangjiaba. The locations of these hydropower stations are signed in Figure 4, and the major parameters of these hydropower stations are listed in Table 1.

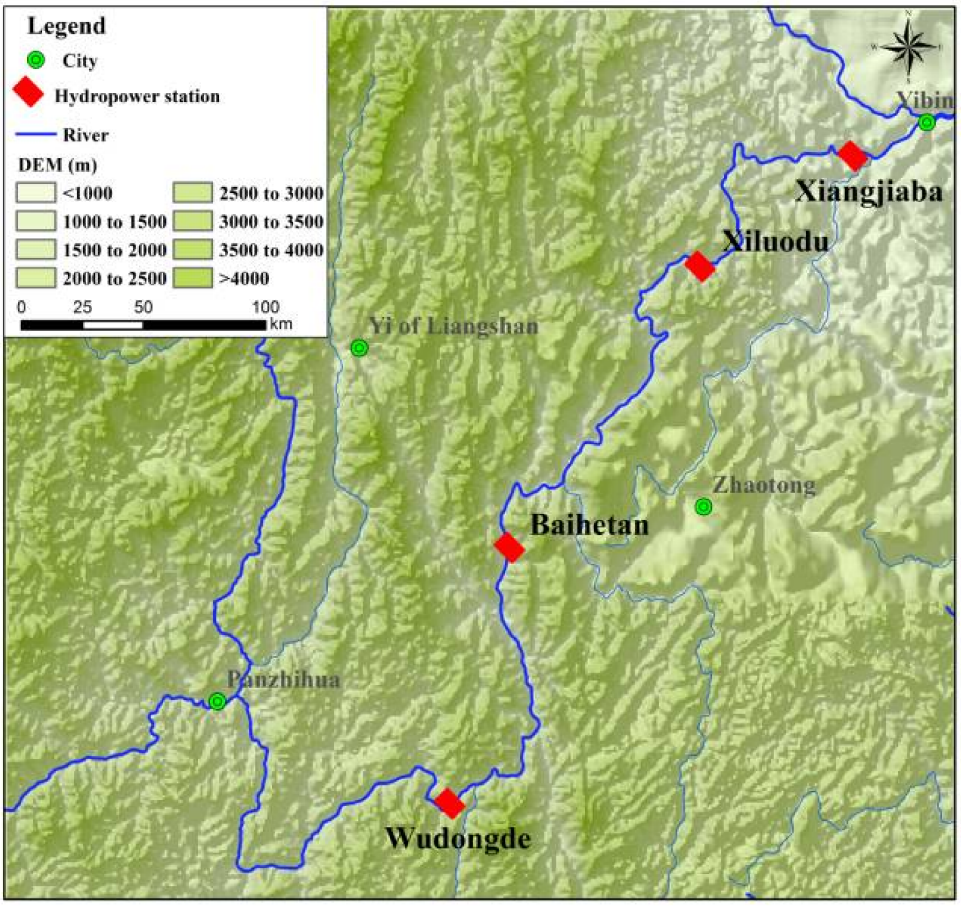


Figure 4. Hydrological stations in the Jinsha River.

Table 1. Main parameters of the four cascade hydropower stations in the Jinsha River.

Parameters	Wudongde	Baihetan	Xiluodu	Xiangjiaba
Dead water level (m)	945	765	540	370
Normal water level (m)	977	825	600	380
Flood limit water level (m)	952	785	560	370
Installed capacity (10^4 kw)	1020	1600	1260	600
Total capacity (10^8 m^3)	74.08	206.27	126.7	51.63
Minimum outflow (m^3/s)	906	905	1500	1500

The aim of long-term hydropower generation scheduling is to determine the water release process of all hydropower stations at each period. The total power generation of different schemes will be

different, thus we need a suitable method to solve this problem. In this case, we do the scheduling simulation of the cascade hydropower stations using the proposed GCS algorithm.

The topology of the four reservoirs in the Jinsha River is shown in Figure 5. The historical observed streamflow of Pingshan hydrological station is used as the input of the LHGS model.

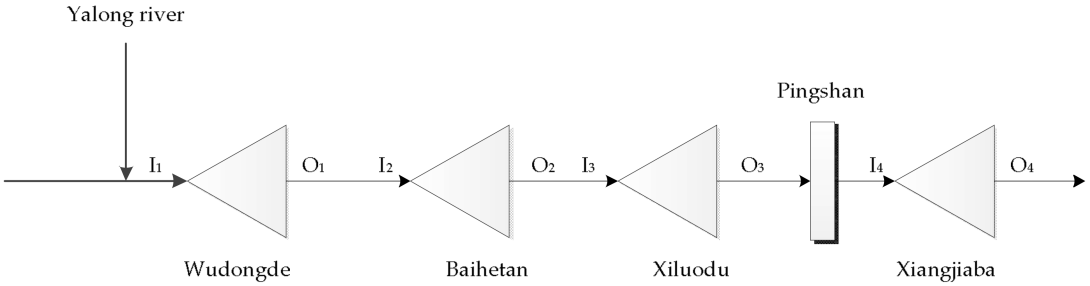


Figure 5. Topology of reservoirs in the Jinsha River.

5.2. Benchmark Function Tests

To evaluate the performance of the proposed ICS, we chose several famous typical benchmark functions, shown in Table 2. Results obtained by proposed ICS algorithms are compared with that simulated by improved harmony search (IHS) [35], the Kbest gravitational search algorithm (KGSA) [36] and basic CS. Of the functions in Table 2, the Ackley function, Griewank function, Rastrigin function, Rosenbrock function and Happy Cat function are multimodal functions. The Sphere function, Bent Cigar function, Discus function and Schwefel 2.22 function are unimodal functions. These functions will be tested under sophisticated features such as shifted and rotated features.

Table 2. Test functions.

Function	Formula	Search Domain	Optimum
Ackley	$f_1 = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + e + 20$	$[-32.768, 32.768]$	$f(0,\cdots,0) = 0$
Griewank	$f_2 = 1 + \frac{1}{4000}\sum_{i=1}^{n}x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$	$[-600, 600]$	$f(0,\cdots,0) = 0$
Rastrigin	$f_3 = An + \sum_{i=1}^{n}\left[x_i^2 - A\cos(2\pi x_i)\right]$	$[-5.12, 5.12]$	$f(0,\cdots,0) = 0$
Rosenbrock	$f_4 = \sum_{i=1}^{n-1}\left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$	$[-5, 10]$	$f(1,\cdots,1) = 0$
Sphere	$f_5 = \sum_{i=1}^{n}x_i^2$	$[-100, 100]$	$f(0,\cdots,0) = 0$
Bent Cigar	$f_6 = x_1^2 + 10^6\sum_{i=2}^{D}x_i^2$	$[-100, 100]$	$f(0,\cdots,0) = 0$
Discus	$f_7 = 10^6x_1^2 + \sum_{i=2}^{D}x_i^2$	$[-100, 100]$	$f(0,\cdots,0) = 0$
Happy Cat	$f_8 = \left\lvert\sum_{i=1}^{D}x_i^2 - D\right\rvert^{1/4} + \left(0.5\sum_{i=1}^{D}x_i^2 + \sum_{i=1}^{D}x_i\right)/D + 0.5$	$[-100, 100]$	$f(-1,\cdots,-1) = 0$
Schwefel 2.22	$f_9 = \sum_{i=1}^{n}\lvert x_i\rvert + \prod_{i=1}^{n}\lvert x_i\rvert$	$[-10, 10]$	$f(0,\cdots,0) = 0$

Dimensions of the benchmark functions are set to 10. The maximum evaluation count is set to 100,000. The default parameters of ICS are $ns = 30, p_a^s = 0.3,\ p_a^e = 0.1,\ sl = 0.01,\ u = 0,\ c = 1.5$. The parameters of IHS and GSA are set to the default values in [35,36]. All of the experiments are performed using a computer with 2.7 GHz Intel i7-4800MQ with 8 GB RAM. The source code is compiled with Java SE8.

Table 3 shows the results of the algorithms including means and standard deviation based on 100 independent tests for each benchmark function. As demonstrated in Table 3, ICS performs superior to IHS, KGSA and CS in most functions except the shifted Rosenbrock function, shifted Happy Cat function, and rotated and shifted Ackley function. Especially on the Sphere, Bent Cigar and Discus, the means and standard deviations of ICS are extremely close to the optimal solutions, much better than other algorithms. While on Rosenbrock function, shifted Happy Cat function, rotated and shifted Ackley function, KGSA performs slightly better than ICS. Overall, ICS can find a satisfactory solution, meaning that the improvement is effective.

Table 3. Errors of test results of the algorithms.

Function		IHS	KGSA	CS	ICS
Shifted Ackley	Mean	4.17×10^{-03}	$\mathbf{6.41 \times 10^{-15}}$ *	1.16×10^{-02}	7.53×10^{-15}
	Stdv	1.28×10^{-03}	$\mathbf{5.54 \times 10^{-15}}$	1.16×10^{-01}	7.13×10^{-15}
Shifted Griewank	Mean	2.34×10^{-02}	1.50×10^{-02}	1.44×10^{-01}	$\mathbf{4.04 \times 10^{-05}}$
	Stdv	3.36×10^{-02}	2.01×10^{-02}	1.32×10^{-01}	$\mathbf{1.85 \times 10^{-04}}$
Shifted Rastrigin	Mean	4.88×10^{-05}	5.89	4.41	$\mathbf{1.00 \times 10^{-06}}$
	Stdv	3.03×10^{-05}	3.51	2.46	$\mathbf{9.84 \times 10^{-06}}$
Shifted Rosenbrock	Mean	2.17	$\mathbf{9.88 \times 10^{-01}}$	2.85	5.68
	Stdv	1.76	$\mathbf{1.75 \times 10^{-01}}$	2.14	1.70
Shifted Sphere	Mean	2.27×10^{-07}	6.88×10^{-32}	2.71×10^{-26}	**0.00**
	Stdv	1.41×10^{-07}	5.78×10^{-31}	9.62×10^{-26}	**0.00**
Shifted Bent Cigar	Mean	$3.48 \times 10^{+01}$	$5.76 \times 10^{+02}$	6.54×10^{-15}	**0.00**
	Stdv	$2.50 \times 10^{+01}$	$8.49 \times 10^{+02}$	7.40×10^{-15}	**0.00**
Shifted Discus	Mean	1.70×10^{-02}	$9.08 \times 10^{+03}$	7.25×10^{-15}	**0.00**
	Stdv	2.74×10^{-02}	$3.27 \times 10^{+03}$	7.14×10^{-15}	**0.00**
Shifted Happy Cat	Mean	1.73×10^{-01}	$\mathbf{3.61 \times 10^{-02}}$	3.14×10^{-01}	1.58×10^{-01}
	Stdv	4.17×10^{-02}	$\mathbf{1.63 \times 10^{-02}}$	1.67×10^{-01}	3.67×10^{-02}
Shifted Schwefel 2.22	Mean	2.03×10^{-03}	4.51×10^{-13}	5.48×10^{-13}	$\mathbf{1.11 \times 10^{-14}}$
	Stdv	5.72×10^{-04}	2.35×10^{-12}	2.97×10^{-12}	$\mathbf{5.92 \times 10^{-15}}$
Rotated and Shifted Sphere	Mean	2.25×10^{-07}	1.63×10^{-01}	8.81×10^{-15}	**0.00**
	Stdv	1.59×10^{-07}	9.49×10^{-01}	6.93×10^{-15}	**0.00**
Rotated and Shifted Ackley	Mean	7.33×10^{-01}	$\mathbf{6.66 \times 10^{-15}}$	2.53	4.22×10^{-12}
	Stdv	1.03	$\mathbf{6.89 \times 10^{-15}}$	1.08	4.21×10^{-11}

* PS: The number in bold represent the minimum value.

5.3. Reservoir Scheduling in Wet Years

To evaluate the gradient strategy for reservoir scheduling, the optimal scheduling scheme of the cascade hydropower stations were obtained by using GCS and compared with those obtained by IHS, KGSA and CS. The parameter of GCS are $ns = 40$, $p_a^s = 0.3$, $p_a^e = 0.1$, $sl = 0.01$, $u = 0$, $c = 1.5$, and the maximum evaluation times is set to 12,000. The initial and end water levels of the reservoirs are set to their normal levels as shown in Table 1. Each month is divided into three periods of approximately 10 days each, resulting in 36 time intervals in a year. Each method is run 100 times, independently.

To evaluate the performance of GCS on reservoir scheduling in wet years, the streamflow of Pingshan station in Figure 6 was chosen as the input of the LHGS model. As shown in Equation (15), the water levels of the reservoirs at different time intervals are the optimization parameters. Because the water level at the 36th time interval is set to the normal water level, the optimization parameters of each reservoir are 35 and the number of reservoirs is 4, meaning that it is a 140-dimensional problem. Equation (1) is the fitness function of LHGS, and the larger power generation means a better scheduling solution.

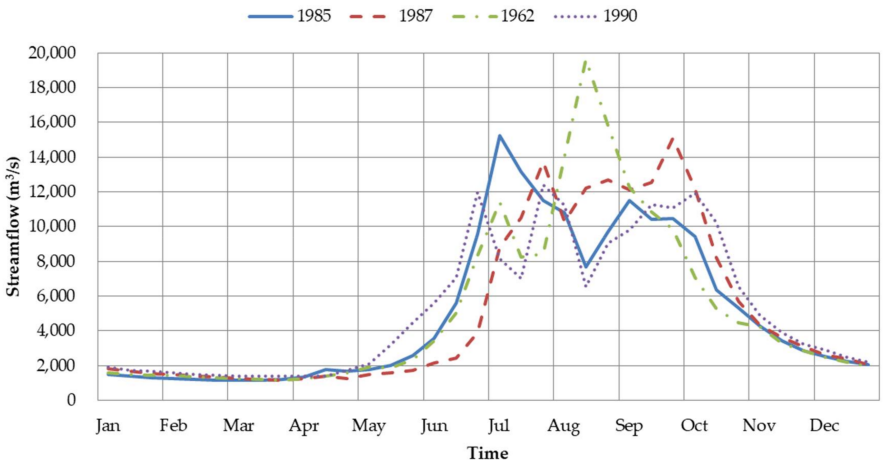


Figure 6. Streamflow of Pingshan station in wet years.

Table 4 clearly shows that GCS is superior to the other three methods. The average power generation obtained by GCS is 232.2, 223.7, 226.3 and 248.4 billion kWh in the four wet years. Compared with IHS, KGSA and CS, the average increases are 4.2%, 8.6% and 2.6%, respectively. It is also noted that the standard deviation obtained by GCS is at least 10 times smaller than that obtained by the other three methods, thus indicating GCS is more stable. Figure 7 shows the convergence process of the four algorithms in the case of the year 1987 as an example. It can be seen from Figure 7 that GCS has the fastest convergence speed, IHS and ICS have not converged yet, while KGSA has fallen into a local optimum.

Table 4. Comparison of power generation (10^8 kWh) in wet years.

Algorithm	1985		1987		1962		1990	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
IHS	2216	27.94	2158	34.23	2171	39.95	2382	40.66
KGSA	2135	12.48	2067	12.45	2076	9.70	2292	12.54
CS	2250	10.77	2189	3.82	2214	5.33	2421	10.34
GCS	2322	0.52	2237	0.02	2263	0.17	2484	0.52

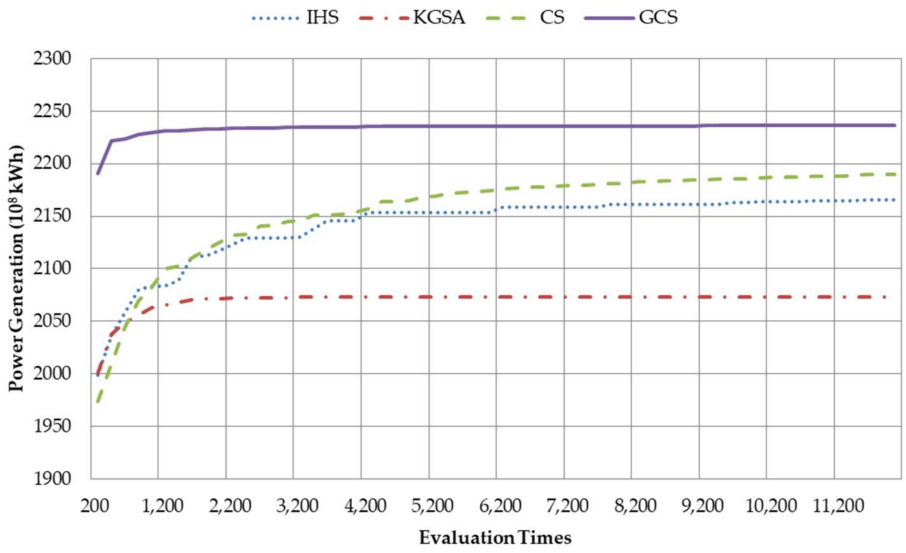
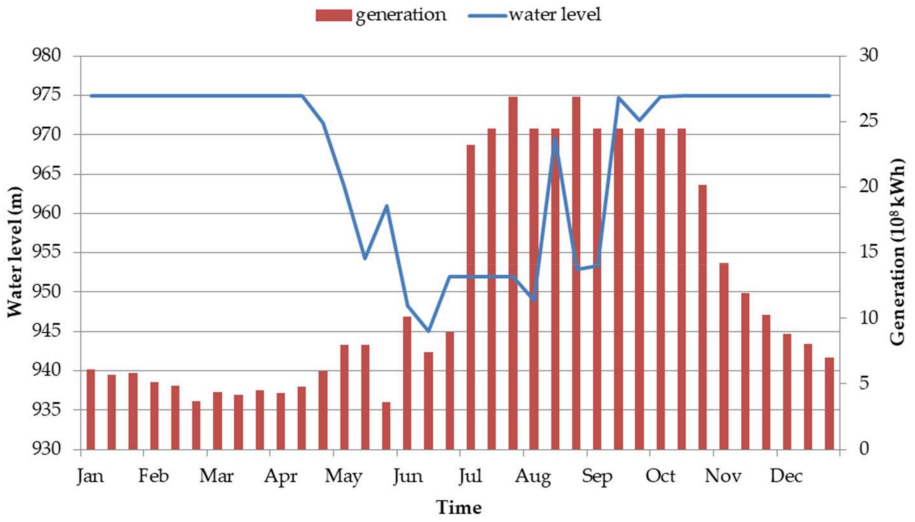
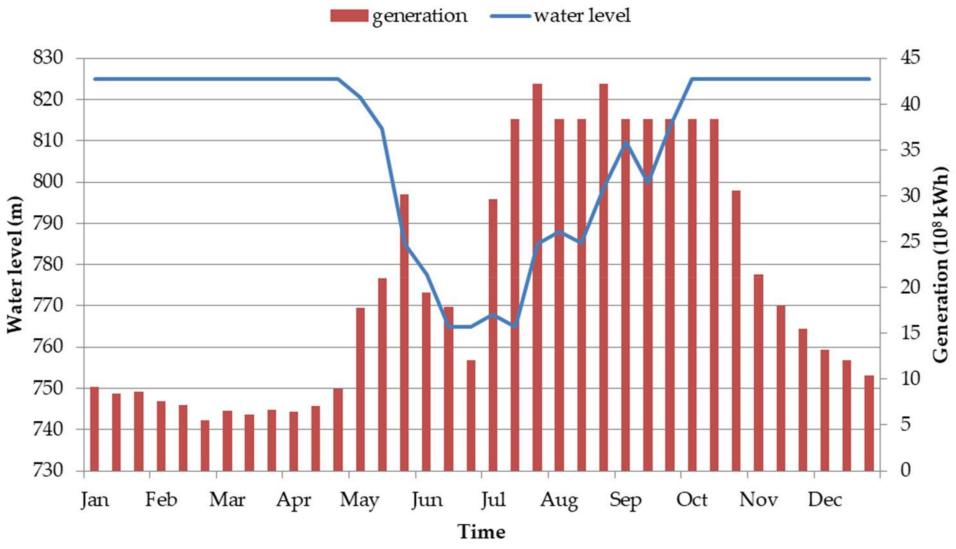


Figure 7. Convergence process of the algorithms in the year 1987.

Figure 8 shows the optimal results of the four hydropower stations in the year 1987. As shown in Figure 6, the streamflow in 1987 increased significantly in July, August and September. The water levels also dropped below the flood limit water level during the flooding season. In the non-flooding season, the water levels remained at the normal level from November to March. The fluctuation of the water level obtained by GCS is small, indicating that it is an effective way to solve LHGS.

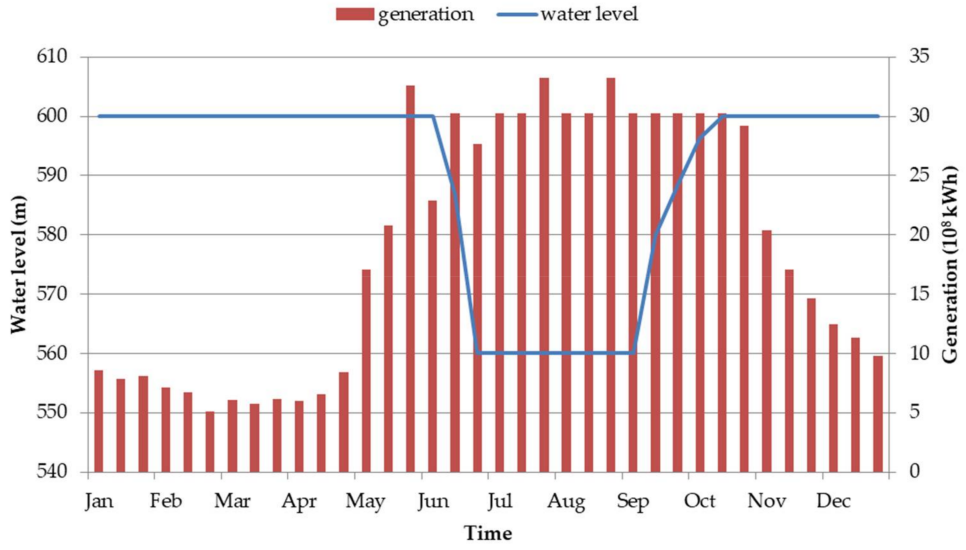


(**a**)

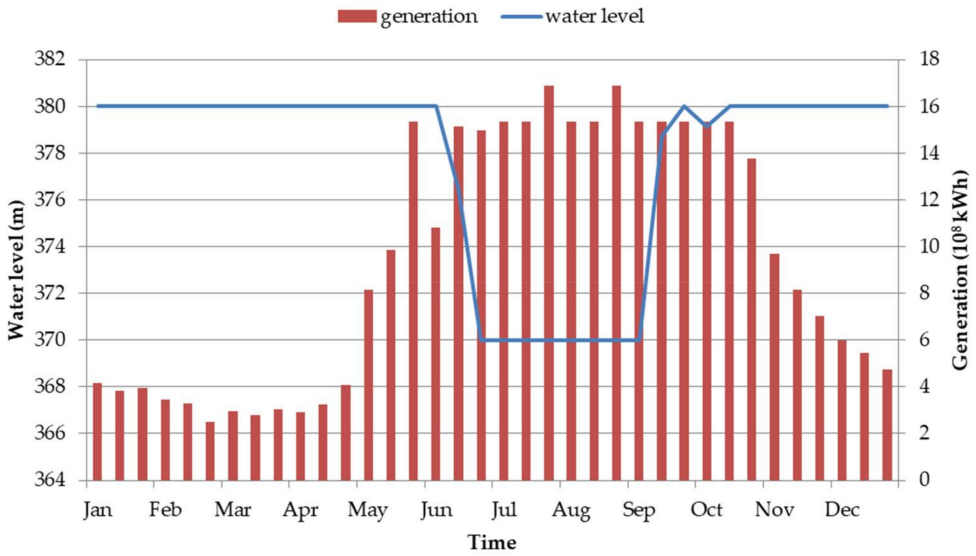


(**b**)

Figure 8. *Cont.*



(**c**)



(**d**)

Figure 8. The optimal results obtained by GCS in wet years: (**a**) Wudongde; (**b**) Baihetan; (**c**) Xiluodu; and (**d**) Xiangjiaba.

5.4. Reservoir Scheduling in Normal Years

In this section, we evaluate the performance of GCS in normal years. Streamflows of the Pingshan station in Figure 9 were chosen as the input of the LHGS model. The parameters and other conditions were the same as those in Section 5.3.

As shown in Table 5, GCS gains more power generation than the other three methods. The average power generation obtained by GCS is 210.5, 220.5, 218.1 and 215.9 billion kWh in the four normal years. Compared with IHS, KGSA and CS, the average increases are 5.8%, 8.6% and 3.3%, respectively. The standard deviation obtained by GCS is also smaller than that obtained by the other three methods. Figure 10 shows the convergence process of the four algorithms in the case of the year 1997 as an example. It can be seen from Figure 10 that GCS has the fastest convergence speed. As in Section 5.3, KGSA does not perform well compared with the other methods.

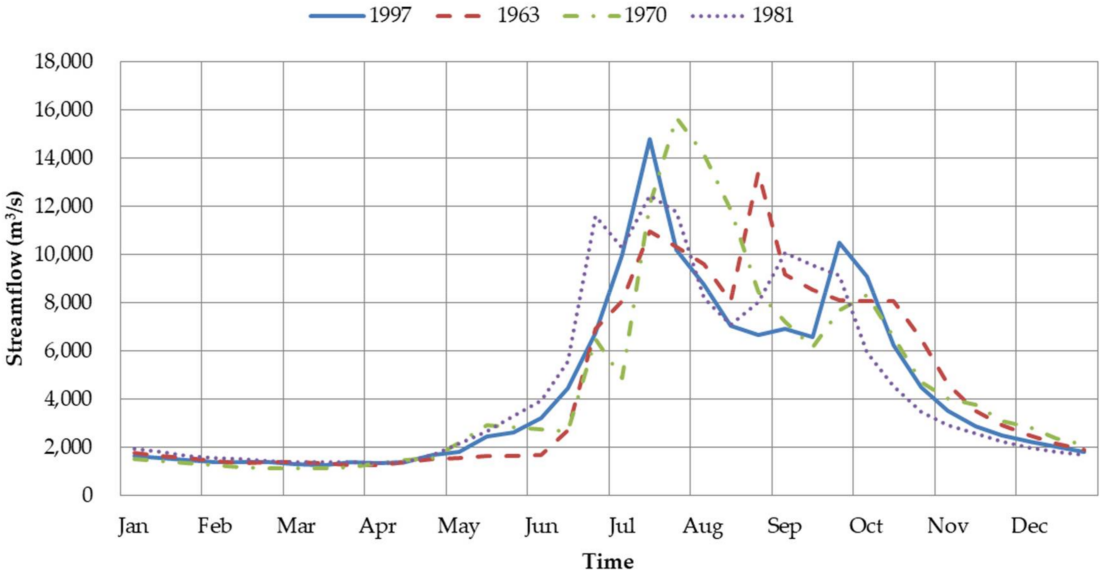


Figure 9. Streamflow of Pingshan station in normal years.

Table 5. Comparison of power generation (10^8 kWh) in normal years.

Algorithm	1997		1963		1970		1981	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
IHS	1966	284.55	2107	29.49	2037	208.99	2065	35.92
KGSA	1953	8.57	2021	204.31	1984	10.67	2007	10.42
CS	2039	5.24	2140	11.24	2096	12.60	2098	6.48
GCS	2105	0.38	2228	0.44	2181	0.13	2159	0.37

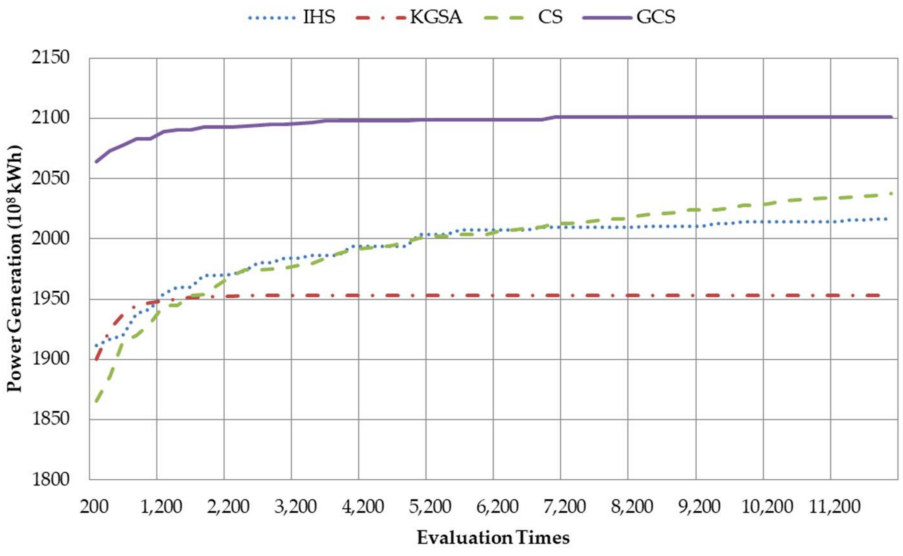


Figure 10. Convergence process of the algorithms in the year 1987.

5.5. Reservoir Scheduling in Dry Years

In this section, we evaluate the performance of GCS in dry years. The streamflows of Pingshan station in Figure 11 were chosen as the input of the LHGS model. The parameters and other conditions were also the same as those in Section 5.3.

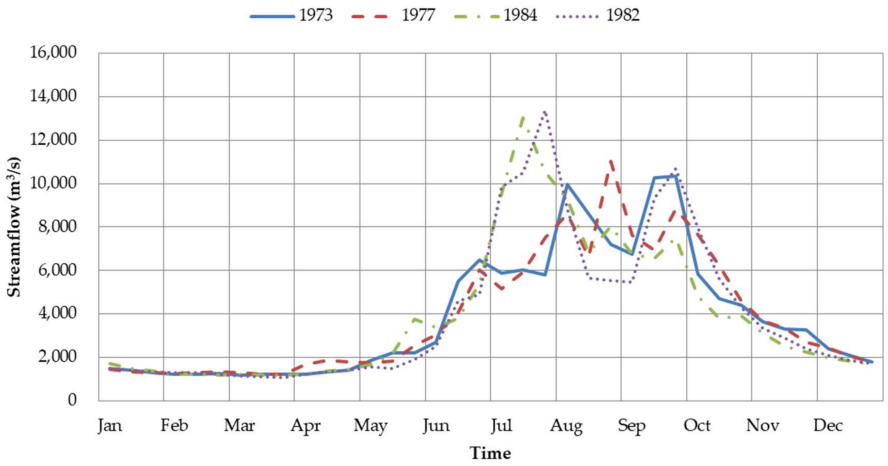


Figure 11. Streamflow of Pingshan station in dry years.

As in Sections 5.3 and 5.4, GCS performed best in these four methods. The mean values obtained by KGSA are the smallest, while the standard deviations obtained by IHS are the largest. As shown in Table 6, the average power generation obtained by GCS is 199.2, 201.9, 194.4 and 197.7 billion kWh in the four normal years. Compared with IHS, KGSA and CS, the average increases are 5.0%, 7.7% and 3.2%, respectively. Figure 12 also shows that the convergence speed of GCS is the fastest. From Sections 5.3–5.5, it is clear that GCS can effectively obtain satisfactory results in different streamflow input scenarios.

Table 6. Comparison of power generation (10^8 kWh) in dry years.

Algorithm	1973		1977		1984		1982	
	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
IHS	1893	37.97	1913	33.33	1862	31.05	1886	38.38
KGSA	1851	6.98	1867	6.63	1809	6.63	1838	6.74
CS	1928	6.09	1946	7.73	1894	4.48	1919	4.49
GCS	1992	0.21	2019	0.07	1944	0.16	1977	0.17

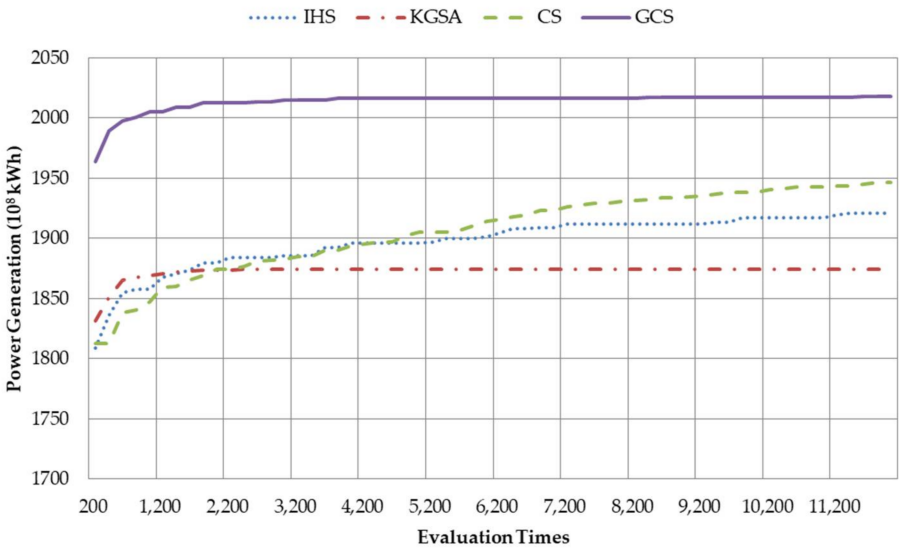


Figure 12. Convergence process of the algorithms in year 1977.

6. Conclusions

The long-term hydropower generation of cascade reservoirs is one of the active research areas in water resources management. According to the operating characteristics of the reservoirs, we propose a gradient strategy to improve the performance of the cuckoo search algorithm when dealing with the LHGS problem. Moreover, the original CS algorithm is adapted with four improvements: a dynamic parameter adjustment of the parameter, a boundary value perturbation strategy for boundary value adjustment, a differential strategy for Lévy flight, and a different way to update solutions.

In the experimental work, the performance of the improvement is shown by several benchmark tests. Results show that the improved CS achieves better results than the other three methods. Finally, the improved CS is hybridized with the gradient strategy and is applied to the LHGS problem in the Jinsha River. The case study shows that the simulation results obtained by GCS are more reliable and stable than those obtained by IHS, KGSA and CS. Compared with IHS, KGSA and CS, the average increases are 4.2%, 8.6% and 2.6% in wet years, 5.8%, 8.6% and 3.3% in normal years, and 5.0%, 7.7% and 3.2% in dry years. The standard deviation obtained by GCS is smaller than that obtained by the other three methods. Overall, GCS can effectively obtain satisfactory results in different streamflow input scenarios.

However, simulation results show that ICS still has drawbacks such as local convergence when dealing with some functions. The uncertainty in reservoir generation scheduling is also not considered yet. Further research is underway to solve these problems.

Acknowledgments: This work is supported by the National Key R&D Program of China (2016YFC0402205), the Key Program of the Major Research Plan of the National Natural Science Foundation of China (No. 91547208), the National Natural Science Foundation of China (No. 51479075, No. 51409008), and the National Public Research Institutes for Basic R&D Operating Expenses Special Project (no. CKSF2017008/SZ). We also acknowledge the entire development team, without whose help this research could not have been undertaken.

Author Contributions: Yu Feng wrote the ICS and GCS algorithm. Chao Wang and Li Mo wrote the LHGS model. Jiang Wu, Zhe Yuan performed the experiments. Final checks were done by Jianzhong Zhou.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Reddy, S.S.; Momoh, J.A. Realistic and transparent optimum scheduling strategy for hybrid power system. *IEEE Trans. Smart Grid* **2015**, *6*, 3114–3125. [CrossRef]
2. Reddy, S.S.; Abhyankar, A.R.; Bijwe, P.R. Market clearing for a wind-thermal power system incorporating wind generation and load forecast uncertainties. In Proceedings of the 2012 IEEE Power and Energy Society General Meeting, San Diego, CA, USA, 22–26 July 2012; pp. 1–8.
3. Reddy, S.S.; Panigrahi, B.K.; Kundu, R.; Mukherjee, R.; Debchoudhury, S. Energy and spinning reserve scheduling for a wind-thermal power system using cma-es with mean learning technique. *Int. J. Electr. Power Energy Syst.* **2013**, *53*, 113–122. [CrossRef]
4. Reddy, S.S.; Bijwe, P.R.; Abhyankar, A.R. Real-time economic dispatch considering renewable power generation variability and uncertainty over scheduling period. *IEEE Syst. J.* **2015**, *9*, 1440–1451. [CrossRef]
5. Liao, X.; Zhou, J.; Ouyang, S.; Zhang, R.; Zhang, Y. An adaptive chaotic artificial bee colony algorithm for short-term hydrothermal generation scheduling. *Int. J. Electr. Power Energy Syst.* **2013**, *53*, 34–42. [CrossRef]
6. Tian, H.; Yuan, X.; Ji, B.; Chen, Z. Multi-objective optimization of short-term hydrothermal scheduling using non-dominated sorting gravitational search algorithm with chaotic mutation. *Energy Convers. Manag.* **2014**, *81*, 504–519. [CrossRef]
7. Wang, K.W.; Chang, L.C.; Chang, F.J. Multi-tier interactive genetic algorithms for the optimization of long-term reservoir operation. *Adv. Water Resour.* **2011**, *34*, 1343–1351. [CrossRef]
8. Zhang, H.; Zhou, J.; Fang, N.; Zhang, R.; Zhang, Y. An efficient multi-objective adaptive differential evolution with chaotic neuron network and its application on long-term hydropower operation with considering ecological environment problem. *Int. J. Electr. Power Energy Syst.* **2013**, *45*, 60–70. [CrossRef]
9. Liao, X.; Zhou, J.; Zhang, R.; Zhang, Y. An adaptive artificial bee colony algorithm for long-term economic dispatch in cascaded hydropower systems. *Int. J. Electr. Power Energy Syst.* **2012**, *43*, 1340–1345. [CrossRef]

10. Soares, S.; Ohishi, T.; Cicogna, M.; Arce, A. Dynamic Dispatch of Hydro Generating Units. In Proceedings of the 2003 IEEE Bologna Power Tech Conference, Bologna, Italy, 23–26 June 2003; Volume 2, p. 6.
11. Cheng, C.T.; Liao, S.L.; Tang, Z.T.; Zhao, M.Y. Comparison of particle swarm optimization and dynamic programming for large scale hydro unit load dispatch. *Energy Convers. Manag.* **2009**, *50*, 3007–3014. [CrossRef]
12. Tu, M.-Y.; Hsu, N.-S.; Yeh, W.W.-G. Optimization of reservoir management and operation with hedging rules. *J. Water Resour. Plan. Manag.* **2003**, *129*, 86–97. [CrossRef]
13. Juhwan, Y. Maximization of hydropower generation through the application of a linear programming model. *J. Hydrol.* **2009**, *376*, 182–187.
14. Nanda, J.; Bijwe, P.R. Optimal hydrothermal scheduling with cascaded plants using progressive optimality algorithm. *IEEE Trans. Power Appar. Syst.* **1981**, *PAS-100*, 2093–2099. [CrossRef]
15. Cheng, C.; Shen, J.; Wu, X. Short-term scheduling for large-scale cascaded hydropower systems with multivibration zones of high head. *J. Water Resour. Plan. Manag.* **2011**, *138*, 257–267. [CrossRef]
16. Chow, V.T.; Maidment, D.R.; Tauxe, G.W. Computer time and memory requirements for DP and DDDP in water resource systems analysis. *Water Resour. Res.* **1975**, *11*, 621–628. [CrossRef]
17. Cheng, C.; Wang, S.; Chau, K.W.; Wu, X. Parallel discrete differential dynamic programming for multireservoir operation. *Environ. Model. Softw.* **2014**, *57*, 152–164. [CrossRef]
18. Wang, C.; Zhou, J.; Peng, L.; Liu, Y. Long-term scheduling of large cascade hydropower stations in Jinsha River, China. *Energy Convers. Manag.* **2015**, *90*, 476–487. [CrossRef]
19. Beck, A.T.; Gomes, W.J.D.S. A comparison of deterministic, reliability-based and risk-based structural optimization under uncertainty. *Probab. Eng. Mech.* **2012**, *28*, 18–29. [CrossRef]
20. Yoo, D.G. Improved mine blast algorithm for optimal cost design of water distribution systems. *Eng. Optim.* **2014**, *47*, 1602–1618.
21. Almasi, M.H.; Sadollah, A.; Mounes, S.M.; Karim, M.R. Optimization of a transit services model with a feeder bus and rail system using metaheuristic algorithms. *J. Comput. Civ. Eng.* **2015**, *29*, 04014090. [CrossRef]
22. Moeini, R.; Soltani-Nezhad, M.; Daei, M. Constrained gravitational search algorithm for large scale reservoir operation optimization problem. *Eng. Appl. Artif. Intell.* **2017**, *62*, 222–233. [CrossRef]
23. Bianchi, L.; Dorigo, M.; Gambardella, L.M.; Gutjahr, W.J. A survey on metaheuristics for stochastic combinatorial optimization. *Nat. Comput.* **2009**, *8*, 239–287. [CrossRef]
24. Blum, C.; Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.* **2003**, *35*, 268–308. [CrossRef]
25. Yang, X.S.; Deb, S. Cuckoo search via levy flights. In Proceedings of the 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), Coimbatore, India, 9–11 December 2009; pp. 210–214.
26. Ouyang, P.; Pano, V. Comparative study of de, pso and ga for position domain pid controller tuning. *Algorithms* **2015**, *8*, 697–711. [CrossRef]
27. Rabanal, P.; Rodríguez, I.; Rubio, F. Parallelizing particle swarm optimization in a functional programming environment. *Algorithms* **2014**, *7*, 554–581. [CrossRef]
28. Nguyen, T.T.; Vo, D.N. Modified cuckoo search algorithm for multiobjective short-term hydrothermal scheduling. *Swarm Evol. Comput.* **2017**, *37*, 73–89. [CrossRef]
29. Din, M.; Pal, S.K.; Muttoo, S.K.; Jain, A. Applying cuckoo search for analysis of lfsr based cryptosystem. *Perspect. Sci.* **2016**, *8*, 435–439. [CrossRef]
30. Bhateja, A.K.; Bhateja, A.; Chaudhury, S.; Saxena, P.K. Cryptanalysis of vigenere cipher using cuckoo search. *Appl. Soft Comput.* **2015**, *26*, 315–324. [CrossRef]
31. Naik, M.K.; Panda, R. A novel adaptive cuckoo search algorithm for intrinsic discriminant analysis based face recognition. *Appl. Soft Comput.* **2016**, *38*, 661–675. [CrossRef]
32. Walton, S.; Hassan, O.; Morgan, K.; Brown, M.R. Modified cuckoo search: A new gradient free optimisation algorithm. *Chaos Solitons Fractals* **2011**, *44*, 710–718. [CrossRef]
33. Mlakar, U.; Fister, I., Jr.; Fister, I. Hybrid self-adaptive cuckoo search for global optimization. *Swarm Evol. Comput.* **2016**, *29*, 47–72. [CrossRef]
34. Du, X.; Wang, J.; Jegatheesan, V.; Shi, G. Parameter estimation of activated sludge process based on an improved cuckoo search algorithm. *Bioresour. Technol.* **2017**, *249*, 447–456. [CrossRef] [PubMed]

35. Mahdavi, M.; Fesanghary, M.; Damangir, E. An improved harmony search algorithm for solving optimization problems. *Appl. Math. Comput.* **2007**, *188*, 1567–1579. [CrossRef]
36. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. Gsa: A gravitational search algorithm. *Inform. Sci.* **2009**, *179*, 2232–2248. [CrossRef]