electronics

MDPI

Article

# Reservoir Optimization Scheduling Driven by Knowledge Graphs

**Hailin Tang [1,2,*], Jun Feng [1,2] and Siyuan Zhou [1,2]**

[1] Key Laboratory of Water Big Data Technology of Ministry of Water Resources, Hohai University, Nanjing 211100, China; fengjun@hhu.edu.cn (J.F.); siyuan.zhou@hhu.edu.cn (S.Z.)
[2] College of Computer Science and Software Engineering, Hohai University, Nanjing 211100, China
* Correspondence: 211307040015@hhu.edu.cn

**Abstract:** As global climate change intensifies, the challenges of water scarcity and flood disasters become increasingly severe. This severity makes efficient reservoir scheduling management crucial for the rational utilization of water resources. Due to the diverse topological structures and varying objectives of different watersheds, existing optimization models and algorithms are typically applicable only to specific watershed environments. This specificity results in a "one watershed, one model" limitation. Consequently, optimization of different watersheds usually requires manual reconstruction of models and algorithms. This process is not only time-consuming but also limits the versatility and flexibility of the algorithms. To address this issue, this paper proposes a knowledge graph-driven method for reservoir optimization scheduling. By improving genetic algorithms, this method allows for the automatic construction of optimization models tailored to specific watershed characteristics based on knowledge graphs. This approach reduces the dependency of the optimization model on manual modeling. It also integrates hydrodynamic simulations within the watershed to ensure the effectiveness and practicality of the genetic algorithms. Furthermore, this paper has developed an algorithm that directly converts optimized reservoir outflow into actionable dispatch instructions. This method has been applied in the Pihe River Basin, optimizing flood control and resource management strategies according to different seasonal demands. It demonstrates high flexibility and effectiveness under varying hydrological conditions, significantly enhancing the operational efficiency of reservoir management.

**Keywords:** genetic algorithm; optimization; reservoir operation; knowledge graph

check for updates

**Citation:** Tang, H.; Feng, J.; Zhou, S. Reservoir Optimization Scheduling Driven by Knowledge Graphs. *Electronics* **2024**, *13*, 2283. https://doi.org/10.3390/electronics13122283

Academic Editor: Alberto Fernandez Hilario

Received: 8 May 2024
Revised: 2 June 2024
Accepted: 7 June 2024
Published: 11 June 2024



## 1. Introduction

As global climate change intensifies, the issues of water scarcity and flood disasters are becoming increasingly severe [1]. In this context, effectively managing reservoirs within a watershed to adequately manage water resources and address flood disasters has become a focal point of research in water management [2]. Reservoir scheduling involves formulating timely water storage and release strategies based on predetermined management objectives and operational constraints to optimize the comprehensive utilization of the reservoirs. In practice, reservoir scheduling is typically modeled as a multi-constraint nonlinear optimization model. The process of determining near-optimal outflow rates through optimization algorithms is referred to as reservoir optimization scheduling [3].

In recent years, with advancements in computer technology, heuristic algorithms inspired by biology, physics, and artificial intelligence have seen widespread development and application [4], especially genetic algorithms. Genetic algorithms adhere to the fundamental principle of "natural selection, survival of the fittest". They eliminate less efficient individuals through the selection of superior genes and processes of crossover and mutation and, through numerous iterations, progressively evolve and select the optimal individuals [5]. Compared to other optimization methods, genetic algorithms feature global optimization, strong adaptability, and robustness, making them widely used in the field of reservoir optimization scheduling [6–8]. When applying genetic algorithms to reservoir optimization scheduling, the outflow from each reservoir in the basin is combined and encoded as the genetic sequence of an individual [9]. During the algorithm's iterative process, these individuals are selected, crossed, and mutated to find the optimal scheduling strategy.

Given the urgency and complexity of reservoir scheduling issues, researchers have recently refined and optimized genetic algorithms to enhance their local search capabilities and computational efficiency. Recent advancements include the introduction of a layered adaptive genetic algorithm equipped with an adaptive dynamic control mechanism, which has been shown to increase computational efficiency through automatic parameter adjustments [10]. Moreover, the transition from binary to decimal representation in genetic coding has been implemented, effectively diminishing the memory demands of these algorithms [11]. The integration of parallel implementations into genetic algorithms has also been observed, markedly boosting their convergence speed and optimization capacity through real-time strategy adjustments [12]. The innovative merging of the bee algorithm with traditional genetic algorithms has led to the introduction of "queen bee" and "alien populations" concepts, which have facilitated the attainment of superior solutions within the same number of iterations [13]. Enhancements in local search capabilities have been achieved by incorporating self-organizing maps into genetic algorithms [14]. Additionally, the optimization of initial population generation, utilizing a chaos model based on Logistic mapping, has been developed to circumvent the pitfalls of local optima, thus improving solution viability [15].

However, despite these improvements, genetic algorithms still exhibit certain limitations in practical applications. Firstly, the diversity in watershed topology and management objectives makes it difficult to broadly apply optimization models and algorithms built for specific watersheds [16]. This specificity leads to a "one watershed, one model" situation [17]. For different watersheds, experts often spend considerable time reconstructing models and algorithms. This process is not only time-consuming but also restricts the versatility and flexibility of the algorithms. Specifically, the interaction among parameters within a watershed depends on its topological structure. Differences in topology between watersheds mean that relationships and constraints among parameters will vary, necessitating customized model construction for each watershed [18]. Additionally, watersheds face different optimization objectives in different seasons. During the flood season, flood prevention is the primary focus. During the non-flood season, the focus shifts to electricity production and water resource management [19]. Second, in reservoir scheduling practices, the outflow rates calculated by genetic algorithms cannot be directly used for operational control of reservoirs. They only serve as auxiliary tools. Dispatchers use these tools to create specific instructions for opening or closing gates based on trends in outflow rates [20]. This process may result in information loss and discrepancies between the effect of dispatch commands and the algorithmically calculated outflow rates. Consequently, there is an urgent need for a universal method that can be applied to different watersheds, automatically construct and solve optimization models, and generate specific dispatch instructions.

Knowledge graph technology represents a cutting-edge method for organizing and processing information [21]. It effectively encodes expert experience and knowledge, organizing real-world entities and their interrelations through a graphical structure. In the field of hydrology, this technology has shown its potential [22]. For example, a knowledge graph developed for the Yellow River Basin, utilizing a comprehensive data repository from the Yellow River Conservancy Commission, has provided robust decision support [23]. Furthermore, the creation of water information knowledge graphs from existing relational databases has underscored the practical applications of this technology [24]. Additionally, the implementation of the Smart Water Management System, which integrates diverse data sources, facilitates multidimensional querying of various water-related data, significantly improving information interconnectivity [25]. By constructing knowledge graphs for watersheds, it becomes possible to elucidate the complex interrelationships and interactions among various water entities. This method supports real-time updates of critical data such as water levels and flow changes, providing essential support for decision making and data-driven management. Consequently, integrating knowledge graph technology with genetic algorithms significantly enhances the responsiveness and predictive accuracy of hydrological models.

Based on the issues above, this paper presents a knowledge graph-driven method for optimizing reservoir scheduling, as follows:

(1) This paper proposes an innovative genetic algorithm utilizing knowledge graphs to build optimization models automatically. It also incorporates hydrodynamic simulation into the genetic operations, enhancing the accuracy and practical value of outflow rates. Additionally, the algorithm demonstrates good adaptability and is suitable for various watersheds without reconstructing models and algorithms. It only requires the construction of a corresponding watershed knowledge graph. Therefore, this algorithm addresses the issue of dependency on manual modelling in constructing watershed optimization models, thereby reducing the cost and complexity of adapting the model to new environments.

(2) This paper has developed an algorithm for generating scheduling commands. This algorithm can directly convert the outflow calculated by the genetic algorithm into actionable commands, such as the gate of the reservoir opening and closing. This enhances the efficiency of translating from theoretical models to practical operations and increases the model's value in practical applications.

(3) This paper applies the proposed algorithm to the Huaihe basin. During the flood season, it implemented targeted single-objective scheduling for flood control, effectively mitigating peak flood levels. This application has also demonstrated that the flood control contributions of reservoirs vary, indicating that not all reservoirs need to be modeled when optimizing for flood control during this period. During the non-flood season, the application considers multi-objective scheduling for power generation and ecological preservation. This demonstrates its broad applicability in complex water resource management tasks.

## 2. Methodology

### 2.1. Watershed Knowledge Graph

This paper introduces knowledge graph technology to model the watershed as a watershed knowledge graph. This graph aims to dynamically reflect time-variant processes in the watershed and provide a rich information resource for genetic algorithms. The graph covers three types of objects: reservoirs, cross-sections, and river channels, as depicted in Figure 1. The left part of the figure displays a generalized watershed diagram, showing Reservoirs *A* and *B*, Sections *F* and *G*, and Channels *C*, *D*, and *E* located between the reservoirs and sections. The right side of the figure presents an instantiated view of these objects and their interrelationships within the watershed knowledge graph. The relationships between cross-sections and reservoirs illustrate the downstream direction of water flow. Additionally, associative links exist between sections, reservoirs, and river channels, which reveal that channels lie intermediate to the associated entities.

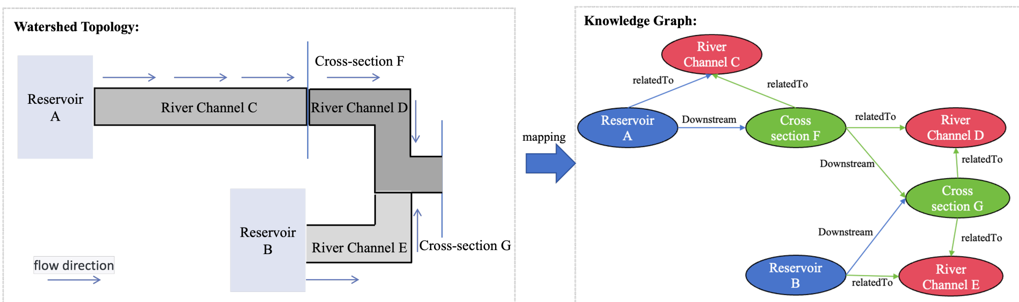


**Figure 1.** Example of the watershed knowledge graph.

The attributes of the reservoirs, cross-sections, and river channels involved in this study are detailed in Table 1. These represent the known quantities prior to the application of genetic algorithms. Attributes can be categorized into three types: static attributes, dynamic attributes, and functional relationships. Static attributes provide unchanging fundamental information about the objects; dynamic attributes describe the time-varying characteristics of the objects based on monitoring and predictive data of the watershed; and functional relationships define the mathematical transformation logic between attributes. It is particularly noteworthy that this paper employs the Muskingum model with consideration of time lag. This model is a widely used method for simulating flood progression, capable of describing the propagation of floods in river channels [26]. Compared to other models, the Muskingum model offers significant advantages in terms of computational complexity and model simplification [27]. Time lag refers to the delay in transmitting the flood from the upstream cross-section to the downstream cross-section of the river channel [28]. This feature allows the model to simulate the flood propagation process more accurately. The Muskingum model is represented as an attribute $\phi$ of the river channel in the watershed knowledge graph. $\phi$ encompasses two critical parameters of the Muskingum model: the flow and weighting coefficients. The flow coefficient indicates the transmission time of the flood within the river channel, while the weighting coefficient reflects the proportion of different water volumes during flood propagation. The algorithm can automatically obtain the Muskingum parameters and compute the flood propagation process through this modeling approach, thereby fully considering the hydrodynamic relationships between upstream and downstream when optimizing reservoir scheduling.

In the classification of reservoir objects, there is a distinction between the leading reservoir located at the uppermost stream and other downstream reservoirs. Critical parameters of the reservoir include water level, inflow, and outflow. The inflow of the leading reservoir is known and usually obtained through hydrological forecasts of the watershed. The outflow of the reservoir is the main objective of the genetic algorithm solution. The water level and inflow of other downstream reservoirs serve as intermediate variables in the solution process, and all are considered parameters to be optimized. The main attributes of the channels include the evolution function and interval inflow. The evolution function calculates the dynamic flow changes within the channel from the inlet to the outlet. Meanwhile, the interval inflow, obtained through hydrological forecasts of the watershed, represents the flow process from the channel to downstream sections or reservoirs. By superimposing the results of the evolution function calculation with the interval inflow, the final flow at the channel outlet of the current tributary can be calculated.

In addition to the three main types of objects previously mentioned, flood discharge structures are modeled, primarily considering the water level–flow relationship and the selectable degree of opening. The control of reservoir outflow depends on several discharge structures, including flood relief tunnels and overflow channels et al., each equipped with multiple operable openings or gates. By adjusting the opening of these structures, the reservoir's outflow can be precisely controlled. Thus, this graph usually associates a reservoir with several discharge structures. The following formula can express the specific control relationship:

$$Q_r^m = \sum_{i=1}^{n} (Q_i \times D_i) \tag{1}$$

where $Q_r^m$ represents the reservoir's discharged flow, and $n$ is the total number of discharge projects. $Q_i$ is the discharge flow of the $i$-th discharge project when fully open. $D_i$ is the degree of openness of the $i$-th discharge project, ranging from 0 (completely closed) to 1 (fully open). For example, if $D_i$ is $\frac{1}{6}$, the structure has six gates or apertures, and only one is open.

**Table 1.** Object categories and their attributes.

| Object Category | Type | Property | Abbreviation | Property Description |
|---|---|---|---|---|
| Reservoir | Static | Name | $NM$ | Official reservoir name. |
| | Static | Design Water Level | $Z_d$ | Highest water level for flood season operation. |
| | Static | Flood Season Limiting Water Level | $Z_{fs}$ | The water level at which regulation begins in flood season. |
| | Static | Ideal Ecological Flow | $Q_{eco}$ | Water flow needed for ecological stability. |
| | Dynamic | Real-time Water Level | $Z_r$ | Current reservoir water level. |
| | Dynamic | Inflow Process | $I_r$ | The future inflow process into the reservoir. (For the upstream-most reservoir) |
| | Function | Water Level–Flow Relationship | $Max$ | Calculate the maximum outflow for a specific water level. |
| | Function | Water Level–Volume Relationship | $d$ | Calculate the maximum storage capacity of the reservoir for a specific water level. |
| River Channel | Static | Name | $NM$ | Official name of the river channel. |
| | Dynamic | Interval Inflow | $\Delta q$ | The flow entering the cross-section or reservoir from outside the river channel. |
| | Function | Channel Evolution | $\phi$ | Calculate the change of flow within the channel. |
| Cross-section | Static | Limiting Flow | $Q_{fs}$ | Section flow rate warning level. |
| | Static | Name | $NM$ | Official section name. |
| Discharge Structure | Static | Name | $NM$ | Official name of the structure. |
| | Static | Optional Opening Degree | $D_i$ | Open portion of the structure. |
| | Function | Water Level–Flow Relationship | $Max$ | Calculate the maximum storage capacity of the reservoir for a specific water level. |

After constructing the watershed knowledge graph, various known parameters within the watershed can be clearly identified. As shown in Figure 2, the inflow $I_r$ to the upstream reservoirs and the interval inflow $\Delta q$ for each river channel are predetermined known parameters. Additionally, although the water level process $Z$ for each reservoir is an intermediate calculation result in the genetic algorithm, the real-time water level $Z_r$, typically starting at $Z_r[0]$, is also considered a known parameter. These defined parameters provide a data foundation for executing the genetic algorithm.

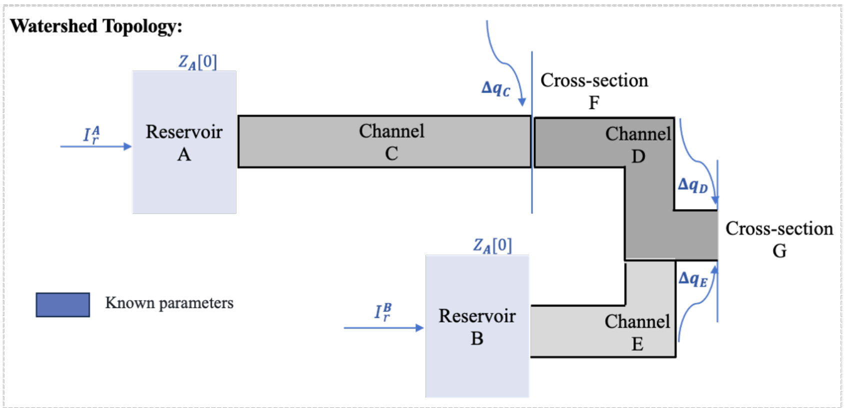


**Figure 2.** Known parameters.

To construct the watershed knowledge graph, we designed an ontology structure based on reservoir scheduling needs. This structure models the hydrological entities within the watershed and their upstream and downstream relationships, including the attributes and relationships of reservoirs, cross-sections, and river channels [23]. In practical applications, entity recognition and relationship extraction can process structured data, like tables, extracting triples that describe the attributes and relationships of hydrological entities. For unstructured data, text analysis techniques based on large language models, using ontology and unstructured text as the Input to generate Cypher statements, can be used to build the knowledge graph. Previously, we validated the effectiveness of this method through experiments. The experimental results indicate that the knowledge graph construction method based on ontology structure and large language models can accurately extract and represent hydrological entities and their relationships within the watershed. It provides rich support information for reservoir scheduling optimization [29].

### *2.2. Knowledge Graph-Driven Genetic Algorithm*

This study proposes a knowledge graph-driven genetic algorithm, which automatically constructs and solves optimization models based on the watershed knowledge graph. Inputs of the algorithm include seasonal types and the watershed knowledge graph. Outputs are the near-optimal outflow process for each reservoir. In the non-flood season, the algorithm considers multiple basin demands such as ecological protection and power generation, implementing multi-objective optimization; during the flood season, it focuses on flood control safety and effective water level management, employing a single-objective optimization strategy. The specific implementation process of the algorithm is as follows:

(1) **Traverse the basin to determine parameters for optimization:** This process involves querying the watershed knowledge graph, identifying the most upstream reservoirs, and randomly selecting one as the starting point for traversal. During traversal, as each new object is visited, its parameters for optimization are identified and recorded. The traversal proceeds downstream along the water flow direction. When a confluence point is reached where multiple objects exist upstream of the current object, the traversal expands to include all upstream tributaries until reaching the uppermost reservoir. After completing the traversal of all tributaries, the process continues downstream until reaching the terminal cross-section of the basin. The parameter-setting strategies for various objects are as follows:

- For the uppermost reservoirs, the parameters to be optimized include the water level process $Z$ and the outflow process $Q_r$.
- For other reservoirs, the parameters include the inflow process $I_r$, the water level process $Z$, and the outflow process $Q_r$.
- For cross-sections, the parameter to be optimized is the flow rate $Q$.

As shown in Figure 3, the outflow from each reservoir, denoted as $Q_r$, is a target for calculation by the genetic algorithm. Thus, it is a parameter to be optimized. The water level process $Z$ of the reservoir is an intermediate result of the genetic algorithm's solution process, determined by $Q_r$ and also considered a parameter to be optimized. For cross-sections $F$ and $G$, their flows $Q_F$ and $Q_G$ are also parameters to be optimized.

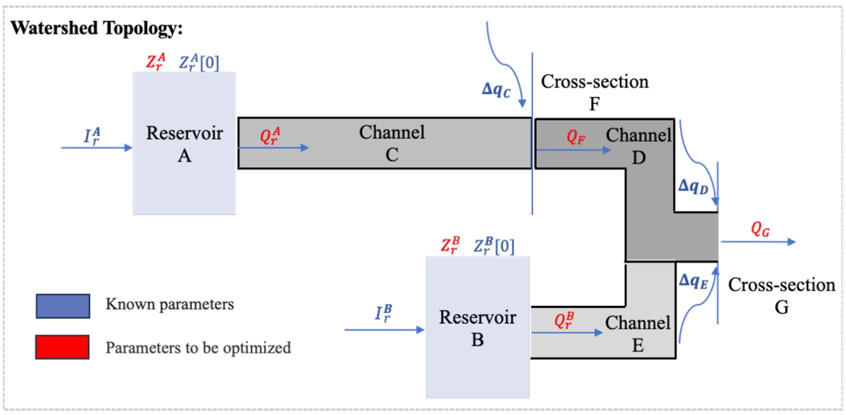


**Figure 3.** Known parameters and parameters to be optimized.

(2) **Initialization of population and Parameters:** In genetic algorithms, a population consists of multiple individuals, each representing a potential solution. The number of genes in each individual is determined by the number of reservoirs and the total data steps.

For example, if there are two reservoirs in the watershed and the total number of data steps is 10, each individual's genetic sequence will include outflow sequences $Q_r$ from the two reservoirs. Each sequence will contain 10 data points, resulting in a total of 20 genes per individual.

Parameters are configured as a dictionary matched to the population size, which stores other parameters that require optimization except outflow. Additionally, this dictionary includes characteristics such as the fitness and congestion of each individual. Specifically, for each individual, population[individual] stores the current individual's outflow sequence $Q_r$ for the reservoirs. Additionally, population[individual][A] contains the current individual's outflow sequence for Reservoir A. Meanwhile, Parameters[individual] includes intermediate calculation results based on the current individual's outflow configuration. This encompasses reservoir inflow $I_r$, water level $Z_r$, and flow $Q$ at watershed sections, as well as characteristics like the individual's fitness and congestion. Thus, there is a one-to-one correspondence between population[individual] and Parameters[individual].

To reduce the complexity caused by hydrological constraints, such as water balance, channel evolution, and water level–flow relationships, all reservoirs are initially set to operate in a single-reservoir scheduling mode during initialization. Single-reservoir scheduling involves each reservoir independently managing and scheduling its water without considering downstream flood control tasks. This mode effectively meets primary water resource distribution and flood prevention requirements. However, single-reservoir scheduling may struggle to effectively manage peak flows during extreme hydrological events such as floods. In such cases, a shift to multi-reservoir joint scheduling strategies is necessary. Multi-reservoir joint scheduling coordinates the operation of various reservoirs to optimize water resource allocation and flood management across the entire watershed. This approach more effectively controls flood risks and ensures water security.

During initialization, an individual that conforms to hydrological constraints is first generated based on the single-reservoir scheduling mode and known parameters. Subsequently, this individual is replicated to the preset population size, typically set at 100. Since all individuals are identical during the initial generation of the population, the intermediate calculation results for each individual remain consistent in the Parameters dictionary. This approach not only simplifies the construction process of the initial population but also ensures the homogeneity of all individuals at startup.

(3) **Choose the appropriate optimization strategy based on the season:** During the flood season, single-objective optimization is implemented. The population is updated iteratively through crossover mutation and fitness-based tournament selection until termination conditions are met, as follows:

- **Crossover mutation**: First, the fittest individual is retained from the population for elite protection. For the rest, a without-replacement sampling method is used to perform crossover mutation operations randomly. This process combines genes from different individuals to create new ones, thereby increasing the population's genetic diversity.
  Whenever an individual's genetic sequence changes, meaning the outflow from the reservoirs included in the individual changes, this necessitates an update. Accordingly, the corresponding intermediate calculation results in the Parameters dictionary must be updated synchronously.
  This study has improved the traditional genetic algorithm's crossover and mutation operations by merging them into a continuous step and incorporating hydrological constraints. These modifications will be detailed in subsequent sections.
- **Fitness calculation**: Since the initial population is produced by cloning, individuals have a high degree of homogeneity. Therefore, the crossover mutation step must first introduce the necessary genetic diversity. Subsequently, fitness calculations are performed, where an individual's fitness is determined based on its performance under a specific optimization function. After the fitness calculation, the fitness value of each individual is updated in the corresponding entry in the Parameters dictionary.

The fitness calculation module defines the paths for retrieving optimization parameters from Parameters or Population for each optimization function. It also sets the formulas for calculating the objective values based on these parameters. To update or add optimization functions, one needs to adjust the paths for retrieving parameters and modify the calculation formulas, thus ensuring the flexibility of the computation process and the scalability of the module.

- **Fitness calculationWinner determination:** A tournament selection strategy is used, randomly dividing the population into several tournament groups. The individual with the highest fitness is selected as the winner from each group.
- **Winner crossover mutation:** Other individuals in the same tournament undergo crossover mutation with the winner. Consequently, the corresponding intermediate calculation results in Parameters are updated.
- **Termination condition determination:** The optimization process is terminated based on predetermined generational limits, such as the number of iterations. Alternatively, it may end when a predefined fitness level is reached.

During the non-flood season, multi-objective optimization is implemented. After initializing the population, it is replicated, crossover mutation is performed on the replicated population, and then the original and mutated populations are merged. Fast, non-dominated sorting and crowding distance calculations are used to determine each individual's Pareto rank. Selection is based on rank and crowding to form a new population, as follows:

- **Clone the population:** To increase the population's genetic diversity, the initial population is cloned to create a similarly sized group with n individuals.
- **Crossover Mutation:** Perform crossover mutation on the cloned population. First, the fittest individual from the population should be retained for elite protection. Apply without-replacement sampling to the remaining individuals and randomly perform crossover mutation to enhance genetic diversity. This step is the same as during the flood season.
- **Merge Populations:** Merge the crossover-mutated population with the original to form a new population consisting of 2n individuals.
- **Fitness Calculation:** Assess the fitness of the merged population. Unlike the flood season, in the non-flood season, each individual's fitness comprises an array of objective values from multiple optimization functions.
- **Fast Non-Dominated Sorting:** Apply fast non-dominated sorting to rank individuals in the population, determining each individual's Pareto rank in a multi-objective optimization environment. A lower Pareto rank indicates that an individual has superior overall performance across multiple objectives, making it more likely to be chosen as an optimal solution.
- **Crowding Distance Calculation:** Assess the crowding distance of individuals within the population. This metric helps maintain the diversity of solutions and prioritizes individuals with a broader distribution. Individuals with higher crowding distances are relatively isolated in the parameter space, reducing the risk of overconcentration in that area and aiding in the exploration of different regions of the parameter space.
- **Final Selection:** Sort individuals by Pareto rank (from low to high) and crowding distance (from high to low). Select the top n individuals from 2n to form the new generation. In the first round of selection of the initial population, the original first-generation population is usually eliminated. These individuals are set based on the maximum discharge capacity of the reservoirs, leading to the maximum peak flow values in the watershed sections. Any crossover mutation operation may reduce these peak values, resulting in poor performance in fitness evaluations.
- **Termination Condition Determination:** Repeat the above process until the set number of iterations is reached or the fitness level meets the predetermined standard, thereby determining whether to end the optimization process.

(4) **Individual Selection:** During the non-flood season, when events such as insufficient ecology or inadequate power generation are detected, the optimal individual for the corresponding objective is selected from the solution set. At this time, within the Pareto front, priority is given to selecting the individual that performs optimally in the corresponding objectives, such as ecological maintenance or power efficiency. During the flood season, given the urgency of flood control, the selection strategy emphasizes flood prevention effectiveness. Therefore, the individual with the best flood prevention effectiveness is chosen from the current population. This selection aims to maximize the reduction in potential water disaster risks in the face of extreme precipitation events.

The algorithm's workflow is illustrated in Figure 4 and Algorithm 1. The algorithm selects different computational strategies based on the varying seasons.

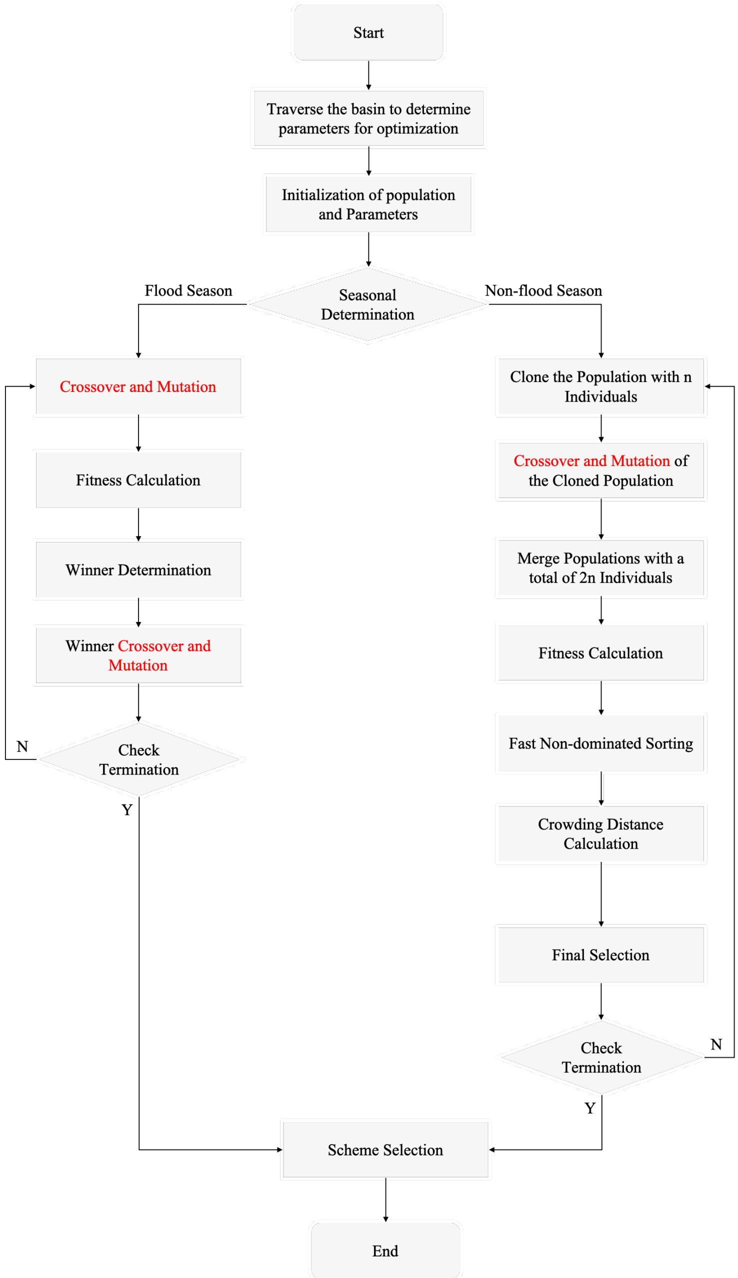


**Figure 4.** Genetic algorithm process.

**Algorithm 1** Knowledge Graph-Driven Genetic Algorithm

```
Input : seasonalTypes, watershed knowledge graph (KG)
Output: near-optimal outflow process Q_r for each reservoir
Initialize:
  population
  Parameters
Begin:
procedure TRAVERSE BASIN
    for each reservoir in KG do
        Parameters[reservoir] ← Extract parameters (Z, Qr) from KG
        Query hydrological and operational constraints from KG for each reservoir
        Store these constraints in Parameters
    end for
end procedure
procedure INITIALIZE POPULATION
    for i ← 1 to population size do
        population[i] ← Generate individual based on single-reservoir mode
        Parameters[i] ← Initialize with Z_r[0], I_r, Q_r from KG
        Update KG with initial conditions of each individual
    end for
end procedure
if seasonalTypes is "flood" then
    SingleObjectiveOptimization(population, Parameters)
else
    MultiObjectiveOptimization(population, Parameters)
end if
while not termination condition do
    CrossoverMutation(population, Parameters)
    Update KG based on crossover results
    FitnessCalculation(population, Parameters)
    Update KG with new fitness values
    population ← TournamentSelection(population, fitness)
    Update KG with selected individuals' details
end while
FinalSelection(population)
Update KG with the final selected outflow processes
Return Q_r from best individual in population
```

### 2.3. Knowledge Graph-Driven Crossover Mutation Module

To ensure that the genetic algorithm accurately reflects and meets the actual hydrological and operational constraints of reservoir outflows, this study proposes an improved crossover mutation strategy. This strategy is based on the topological relationships in the watershed knowledge graph. It merges crossover and mutation operations into a continuous process, incorporates hydrological constraints, and is termed crossover mutation. The main constraints considered in this step are:

- **Channel evolution constraints:** The change in the outflow $Q_r(t)$ from upstream reservoirs as it evolves through river channels to downstream reservoirs or cross-sections at any given time [30].
- **Water balance constraints:** The impact of the outflow $Q_r(t)$, inflow $I_r(t)$, and current water level $Z_r(t)$ of a reservoir on the water level at the next moment, $Z_r(t+1)$ [31].
- **Water level–flow constraints:** The limitations imposed by the water level $Z_r(t)$ of a reservoir on its maximum discharge capacity $Q_{\max}(t)$ at any given moment [32].

The Input to the crossover mutation module includes two individuals from the population, individual1 and individual2, along with the Parameters dictionary. The output consists of the newly created individuals, new_individual1 and new_individual2, and the updated Parameters dictionary. The specific steps of the crossover mutation are as follows:

(1) Current reservoir crossover mutation: First, randomly select a reservoir $i$ within the watershed and determine the crossover time $t$. At this time, perform a crossover operation on the outflow sequences of the two individuals at the specified reservoir and time. After the crossover, the same mutation operation is performed on both individuals. The specific operations are as follows, where "individual" refers to either individual1 or individual2. Based on the exchanged outflow $Q_r(t)$, look up the water level $Z_r(t)$ and inflow $I_r(t)$ of the crossover reservoir at time $t$ in Parameters[individual]. Then, calculate the next moment's water level $Z_r(t+1)$ using the water balance constraint. First, determine the maximum allowable outflow $Q_max(t+1)$ based on the water level–flow relationship. Then, randomly generate the outflow $Q_r(t+1)$ within this range and update it in the genetic sequence. Repeat this process until all time points for that reservoir are updated, and update the new $Z_r(t)$ in Parameters[individual].

(2) Mutate downstream reservoirs: According to the watershed knowledge graph's watershed topology and object types, traverse downstream from the current reservoir. Then, mutation operations on the traversed path are performed. The strategy for mutation operations is as follows:

- When reaching a section, use the channel evolution function of the river channel upstream of the section from the watershed knowledge graph to calculate the flow at the section after time $t$. Combine this channel evolution result with the river's interval inflow to determine the total flow of the current tributary. If no tributaries are upstream of the section, this total flow will be directly updated as the flow $Q$ of the section in Parameters[individual]. If there are tributaries, use a similar method to calculate the total flow of each tributary. Sum these flows to determine the flow $Q$ at the section, then update this new $Q$ in the corresponding individual's Parameters[individual].
- When reaching a reservoir, first apply the same strategy used for updating the section flow $Q$ to update the inflow $I_r(t)$ from time $t$ in Parameters[individual]. Then, check the current water level $Z_r(t)$ and apply the water balance constraint for that reservoir from the watershed knowledge graph. Using the water level–storage relationship, calculate the water level $Z_r(t+1)$ for the next moment. Finally, based on the water level–flow relationship, randomly generate the outflow $Q_r(t+1)$ for the next moment and update this in the individual's genetic sequence. This updating process continues until all steps are completed, and the updated $Z_r$ and $I_r$ are written into Parameters[individual].

After completing the traversal of the watershed's terminal section, the crossover mutation process also concludes. Consequently, the water levels and flow rates of reservoir $i$ and all downstream reservoirs and sections at time $t$ and subsequent periods have been updated. These updates are based on water balance, channel evolution constraints, and water level–flow relationships. These updates accurately reflect the dynamic changes in the watershed after changes in outflow, thus ensuring the effectiveness of the genetic algorithm in reservoir optimization scheduling. Figure 5 illustrates an example of applying the crossover mutation module, where (a) represents the watershed topology and (b) depicts the process of crossover mutation. The numbers in the diagram indicate the order of parameter calculation in the current object. Arrows show the direction of traversal; yellow represents the outflow from the reservoirs in the individual genetic sequences; blue represents intermediate computational results contained in Parameters; and red indicates parameters that are pending update.

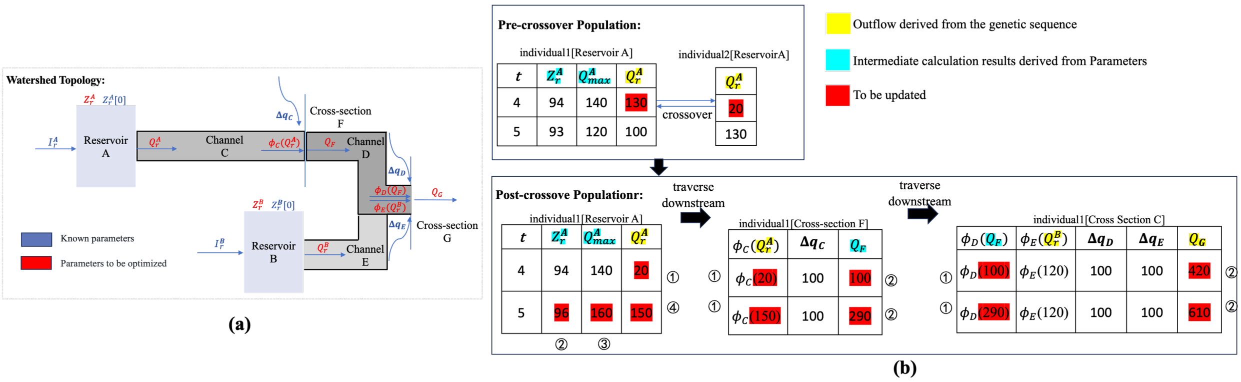


**Figure 5.** Example of the crossover mutation. (**a**) the watershed topology; (**b**) the process of crossover mutation.

As is shown in Algorithm 2, first, the algorithm randomly selects Reservoir $A$ for crossover and sets the crossover time at $t$ = 4. Since the crossover occurs at $t$ = 4, all parameters of Reservoir $A$ and its downstream hydraulic objects before t = 4 do not require updating; only changes at $t$ = 4 and beyond need to be considered. For Reservoir $A$, ① indicates that through crossover with individual2, the outflow of individual1[Reservoir $A$] at $t$ = 4 changes to 20. ② is based on the inflow, water level, and outflow at $t$ = 4, using the water balance constraint to calculate the water level at $t$ = 5, which results in 96m due to reduced outflow and increased water storage in the reservoir. ③ indicates that based on the current water level of 96 m and the water level–flow relationship, the maximum discharge capacity of the reservoir at the current moment is 160 cubic meters per second. ④ states that, with a lower bound of 0 and an upper bound of 160 cubic meters per second, the outflow at the current moment is randomly generated as 150 cubic meters per second. At this point, the crossover and mutation process for Reservoir $A$ is complete, and the procedure moves downward to continue the mutation process.

The algorithm traverses down to Section $F$. ① indicates that based on the latest outflow from Reservoir $F$, the flow channel evolution to Reservoir $B$ is calculated. For ease of calculation, the channel evolution function "$\phi$" is set as a constant 1. Thus, $\phi(20)$ equals 20 cubic meters per second. ② shows that the result of the $\phi$ function is summed with the interval inflow (fixed at 100), yielding the flow process for Section F. The mutation process for Section F is thus completed.

Then, the traversal continues downward to Section $G$. ① indicates that based on the flow at Section $F$, the flow channel evolution to Section $G$ is calculated. ② shows that the result of the $\phi$ function, combined with the interval inflow (fixed at 100 cubic meters per second), yields the total flow for the current tributary. This total is then added to that of another tributary to determine the flow process at Section $G$. Since Section $G$ is the terminal Section, the crossover mutation process concludes here. As the crossover involves Reservoir $A$, there is no need for mutation and updates for Reservoir $B$, which is located on a different tributary.

**Algorithm 2** Knowledge Graph-Driven Crossover Mutation

```
Input: Watershed knowledge graph, individual1, individual2, Parameters
Output: new_individual1, new_individual2, updated Parameters
procedure CROSSOVERMUTATION(individual1, individual2, Parameters)
    Select a reservoir i and a crossover time t randomly
    Perform crossover on outflow sequences Q_r^i(t) of individual1 and individual2 at reservoir i and time t
    Apply mutation operation to both new individuals
    Query initial conditions from the knowledge graph for reservoir i
    for each time t from crossover time to end do
        Z_r^i(t+1) ← Calculate the next water level using water balance constraints from KG
        Q_max^i(t+1) ← Determine max outflow using water level–flow relationship from KG
        Randomly set Q_r^i(t+1) within [0, Q_max^i(t+1)]
        Update Q_r^i(t+1) in new individuals
    end for
    Update Z_r^i(t) in Parameters for both new individuals
    Update KG with new water levels and outflows for reservoir i
    for each downstream object from reservoir i do
        if object is a section then
            Calculate flow at section using channel evolution function from KG
            Sum flows from tributaries as defined in KG
            Update flow Q in Parameters
            Update KG with new flows at section
        else if object is a reservoir then
            Update inflow I_r(t) in Parameters based on new outflows
            Recalculate and update water level Z_r(t+1) using KG
            Update outflow Q_r(t+1) based on new water level
            Update KG with new inflows, water levels, and outflows for the reservoir
        end if
    end for
    After completing traversal, finalize updates in Parameters
    Update KG with final conditions for all objects traversed
end procedure
Return new individuals and updated Parameters
```

### *2.4. Scheduling Instruction Generation Algorithm*

The scheduling instruction generation algorithm inputs the discharge structure opening levels and water level–flow relationships from the watershed knowledge graph and the outflow process of the reservoir. Its outputs are the status and actions of the discharge structures at each moment. Precisely, based on the known outflow $Q_r^m$, the algorithm can calculate and determine the degree of openness for each discharge project. This allows it to set the degree of openness and scheduling actions at each moment in response to water level and flow changes. As is shown in Algorithm 3, the specific steps implemented by the algorithm are as follows:

**(1) Generating Options:** The algorithm first queries the water level–flow relationship and the selectable degrees of openness of discharge structures from the watershed knowledge graph. Subsequently, based on the selectable degrees of openness for each structure, the algorithm generates a series of optional scheduling plans. For example, assuming there are three discharge structures, a, b, and c, where a must be opened, and c provides two open options, the algorithm can generate all possible options, including (a), (a, b), (a, b, c), (a, b, 1/2c), (a, c), and (a, 1/2c).

**(2) Selecting the Best Option:** The algorithm will use all possible options to traverse the time series of water levels and flows, calculating the maximum flow for each option at each time point. First, it will eliminate options where the maximum flow is less than the actual flow. Among the filtered options, the algorithm calculates the gap between the flow generated by each option and the current actual flow, selecting the option with the smallest gap as the optimal openness plan. It then generates a sequence of plans representing the status of each project at each time point. For example, [23 March 2024: (a, b, 1/2c); 24 March 2024: (a, c)].

**(3) Generating Scheduling Actions:** Based on the sequence of plans generated from the analysis above, the algorithm compares the openness of discharge projects between two consecutive time points. By analyzing changes in openness, the algorithm determines the specific scheduling actions needed at the current time. For example, if the plan at time t-1 was (a, b, 1/2c) and changes to (a, c) at time t, it indicates the need to close b and fully open c. Ultimately, a sequence of actions is generated, such as [23 March 2024: (a to 1, b to 1, c to 1/2); 24 March 2024: (b to 0, c to 1)].

**Algorithm 3** Scheduling Instruction Generation Algorithm

**Input:** Watershed knowledge graph, outflow process $Q_r^m$
**Output:** Status and actions of discharge structures
**procedure** GENERATESCHEDULINGINSTRUCTIONS(KnowledgeGraph, $Q_r^m$)
    Query OpennessLevels, WaterFlowRelationships from KnowledgeGraph
    Generate possible openness combinations based on discharge structure levels from KnowledgeGraph
    **for** each time step **do**
        Evaluate all combinations against the outflow $Q_r^m$
        Eliminate combinations where the max flow < actual flow
        Calculate the gap between possible and actual flow
        Select the combination with the smallest gap
        Record the selected combination as the plan for that time
    **end for**
    **for** each transition between consecutive time steps **do**
        Compare the selected combinations
        Determine necessary adjustments in discharge structures
        Generate scheduling actions based on adjustments
    **end for**
    Update KnowledgeGraph with the new discharge structure status and actions
    **Return** Updated KnowledgeGraph, Scheduling Actions
**end procedure**

## 3. Performance Evaluation

### *3.1. Watershed Overview*

This study selects the *Pihe River Basin* as the research watershed. As shown in Figure 6, the core protection object in this watershed is the *Zhengyangguan* section. This area has four main reservoirs: *MoziTan* Reservoir, *Foziling* Reservoir, *Bailianya* Reservoir, and *Xianghongdian* Reservoir. These reservoirs jointly ensure flood safety at *Zhengyangguan*, with *Xianghongdian*, *Foziling*, and *MoziTan* also tasked with protecting downstream channel flood safety [33].

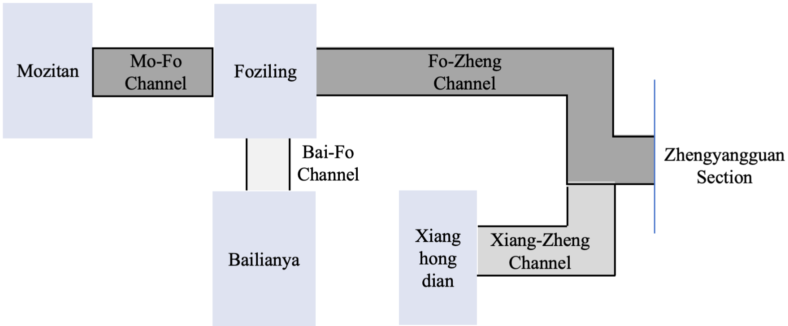


**Figure 6.** Schematic map of the Pihe River Basin.

The main attributes of the four reservoirs within the watershed are shown in Table 2. For *Xianghongdian*, *Foziling* , and *MoziTan* Reservoirs, to simplify the watershed topology, no separate sections are set downstream of these three reservoirs. Instead, the safe discharge capacity of downstream channels is set to the maximum discharge flow during operation, ensuring the safety of the downstream channels.

**Table 2.** Key attributes of the reservoirs.

| Reservoir Name | Total Capacity (Bm³) | Verification Water Level (m) | Design Water Level (m) | Flood Control Capacity (Bm³) | Maximum Discharge Flow (m³/s) | Downstream Channel Safe Discharge (m³/s) |
|---|---|---|---|---|---|---|
| Xianghongdian | 26.10 | 143.37 | 140.98 | 5.00 | 5121 | 1500 |
| Foziling | 4.91 | 129.80 | 125.97 | 0.80 | 7750 | 3450 |
| MoziTan | 3.47 | 203.79 | 197.28 | 1.12 | 4250 | 4000 |
| Bailianya | 4.60 | 234.50 | 209.24 | 2.75 | 5049 | N/A |

The *Pihe River Basin* is modeled as a watershed knowledge graph, including sections, reservoirs, and river channels within the watershed, as illustrated in Figure 7. We collected hydrological data, scheduling rules, historical operation records, and other information for this watershed. For unstructured data, we employed text analysis techniques based on large language models to extract relevant entities and relationships from the text and generate corresponding Cypher statements. For structured data, such as tables and database records, we used entity recognition and relationship extraction techniques to automatically generate triples describing hydrological entities and their relationships. By integrating these structured and unstructured data, we successfully constructed the knowledge graph for the Pihe River Basin, providing a solid foundation for comprehensive management and optimized scheduling of the watershed.

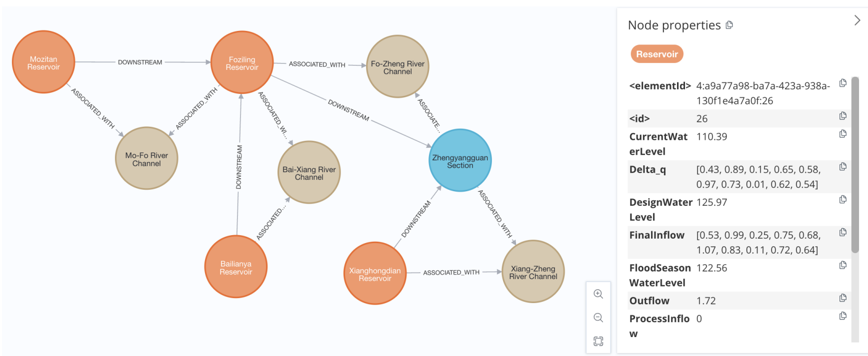


**Figure 7.** Watershed knowledge graph of the Pihe River Basin.

The flood control task of the *Pihe River Basin* during the flood season is to ensure the flood safety of the *Zhengyangguan* section. In the non-flood season, reservoir water levels are generally low due to lower rainfall and smaller inflow. The main objectives considered in such cases are ecological water supply and maximizing power generation [34]. The optimization functions for the flood and non-flood seasons of the watershed are as follows:

**(1) Maximum Peak Reduction (Flood Season):** This objective function is intended to quantify the effect of peak reduction, represented explicitly by minimizing the peak flow at *Zhengyangguan*, thereby ensuring effective control of the peak flow. The flow process at the *Zhengyangguan* section is denoted by $Q_{\text{zheng}}$, and the maximum value in this flow process, i.e., the peak flow, is represented as:

$$f_1 = \min(\max\{Q_{\text{zheng}}\}) \tag{2}$$

where $Q_{\text{zheng}}$ is the flow process at the *Zhengyangguan* section, and $\max\{Q_{\text{zheng}}\}$ is the maximum value in the flow process at *Zhengyangguan*, i.e., the peak flow.

**(2) Maximum Power Generation (Non-Flood Season):** When the outflow from the reservoirs, head, and turbine efficiency are relatively stable, the power generation is directly proportional to the outflow. Considering that a too low flow may not fully utilize the power generation capacity of the power stations, the objective function aims to maximize the total flow of all reservoirs over a given period:

$$f_2 = \max \sum_{n=1}^{N} \sum_{t=1}^{T} Q_r^m(t) \tag{3}$$

where $N$ represents the total number of reservoirs, $T$ is the total number of time periods considered, and $Q_r^m(t)$ is the outflow from the $m$-th reservoir in the watershed during time period $t$.

**(3) Minimum Ecological Water Supply Deviation (Non-Flood Season):** This objective function aims to meet human water and power demands while ensuring the minimum deviation from the required ecological flow in the watershed. The formula is given by:

$$f_3 = \min \sum_{m=1}^{M} \sum_{t=1}^{T} |Q_{\text{eco}}^m - Q_r^m(t)| \tag{4}$$

where $M$ represents the total number of reservoirs; $T$ the total number of time periods considered; $Q_{\text{eco}}^m$ the ideal ecological flow for the $m$-th reservoir; and $Q_r^m(t)$ the outflow from the $m$-th reservoir during time period $t$.

There is an apparent conflict between objectives $f_1$ and $f_2$: higher flows benefit the maximum power generation objective, while lower flows favor the minimum ecological water supply deviation objective. Therefore, a balance must be struck between these two objectives during the scheduling process.

### *3.2. Single-Objective Scheduling during Flood Season*

Single-objective scheduling during the flood season was based on the real-time data collected from the *Pihe River Basin* from 14:00 to 23:00 on 21 July 2020. This dataset includes interval inflows for each channel and inflows to the uppermost *MoziTan*, *Bailianya*, and *Xianghongdian* Reservoirs. Data from ten hours with higher inflows were selected, corresponding to the actual scheduling cycle of the watershed. As shown in Figure 8, (a) represents the inflow process to the upper reservoirs, and (b) shows the interval inflow within each channel of the watershed.

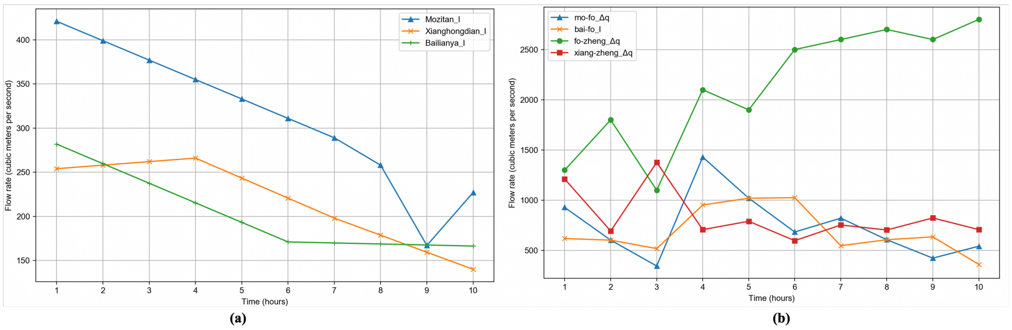


**Figure 8.** Inflows to the upper reservoirs and interval inflow of channels. (**a**) the inflow process to the upper reservoirs; (**b**) the interval inflow within each channel of the watershed.

Initially, this study implemented a strategy of setting all reservoirs within the watershed to single-reservoir scheduling during population initialization. After setting the reservoirs for single-reservoir scheduling, the first generation of the population was generated, with initial water levels at *MoziTan*, *Foziling* , *Xianghongdian*, and *Bailianya* Reservoirs set to 185 m, 115 m, 129.83 m, and 203.37 m, respectively. Each reservoir's outflow was discharged according to the maximum discharge capacity based on the water level–flow relationship, as shown in Figure 9. The population consisted of individuals with 40 genes, comprising outflow data from four reservoirs over ten time steps. During the population generation process, intermediate values were calculated, such as the water level changes of the reservoirs and the flow process at the *Zhengyangguan* section, and added to the Parameters dictionary.

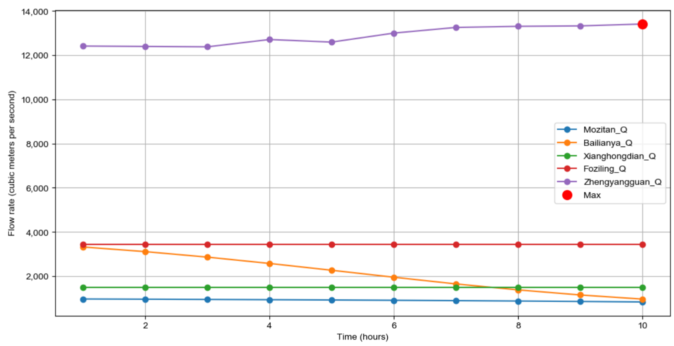


**Figure 9.** Changes in flow within the watershed due to single-reservoir scheduling.

In the genetic algorithm settings, the predetermined number of populations was set to 100, with each population containing 100 individuals. After initializing the population, crossover mutation and selection processes were carried out on the first generation of 100 individuals to optimize their fitness. Through 100 generations of iteration, individuals with higher fitness were retained, and the optimal outflow strategy for each reservoir was ultimately determined.

Due to differences in each reservoir's scale, location, and storage capacity, their contributions to flood protection within a watershed vary [35]. Including all reservoirs in the modeling process might not significantly enhance flood control and would increase the complexity of the model [36]. Related studies in the *Pihe River Basin* have already addressed the issue of selecting reservoirs with significant flood control contributions for scheduling [37]. During floods at *Zhengyangguan*, these key reservoirs should be jointly scheduled and modeled as a single system, while other reservoirs can be managed individually. The outflow rates of these individually scheduled reservoirs will be updated in the watershed knowledge graph and used as inputs for the optimization model. Therefore, to comprehensively evaluate the flood control contributions of the reservoirs, this study designed four different scheduling strategies for comparative experiments: (a) joint scheduling of *Foziling* and *Xianghongdian* Reservoirs; (b) joint scheduling of *Xianghongdian*, *Foziling* , and *Bailianya* Reservoirs; (c) joint scheduling of *Xianghongdian*, *Foziling* , and *MoziTan* Reservoirs; and (d) comprehensive joint scheduling of all the aforementioned reservoirs. These experiments aimed to analyze the impact of different joint scheduling strategies on the flow process at the *Zhengyangguan* section.

The experimental results are displayed in Figure 10, which details the optimal outflow from the reservoirs under different scheduling strategies. Meanwhile, the process of water level changes in the reservoirs is shown in Figure 11. The dashed line represents the design flood level of the respective reservoir, which should not be exceeded during scheduling.

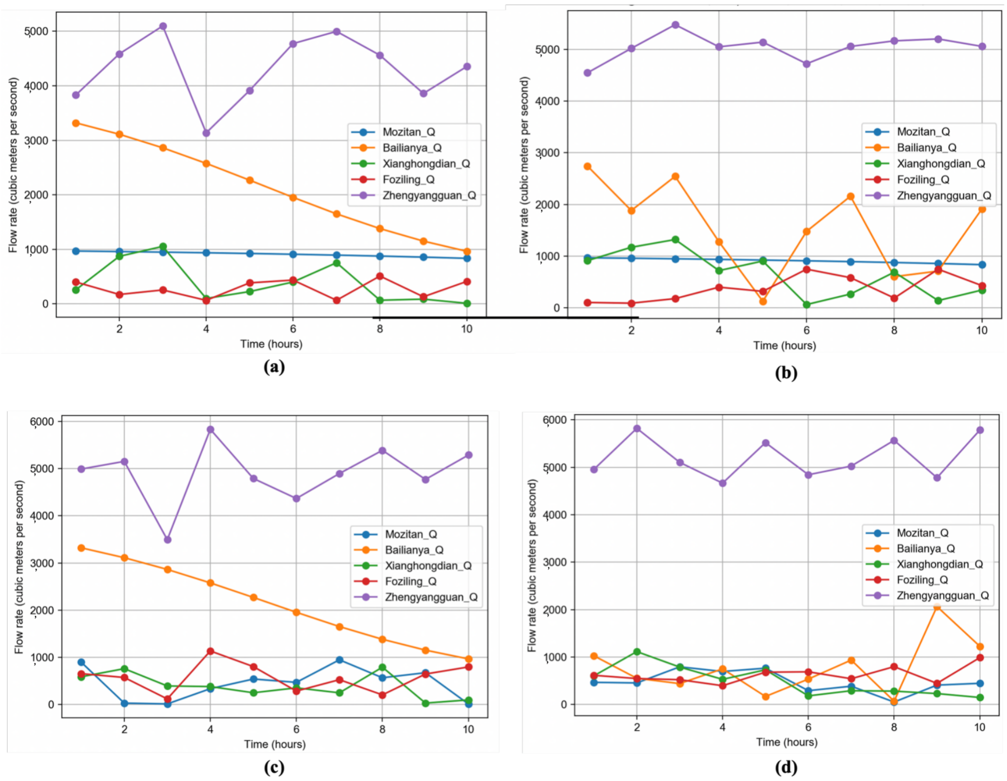


**Figure 10.** Optimal z outflow from reservoirs under each scheduling strategy. (**a**) joint scheduling of Foziling and Xianghongdian Reservoirs; (**b**) joint scheduling of Xianghongdian, Foziling , and Bailianya Reservoirs; (**c**) joint scheduling of Xianghongdian, Foziling, and MoziTan Reservoirs; and (**d**) comprehensive joint scheduling of all the aforementioned reservoirs.

It is evident from the charts that under the four different scheduling strategies, the reservoirs' water levels were within the designed flood levels, indicating that the reservoirs' safety was not severely threatened during the implementation of these scheduling plans, thus ensuring structural safety. Specifically, from Figure 10 , it can be observed that the peak flow rates at the *Zhengyangguan* section under the four strategies were 5097 $m^3/s$, 5478 $m^3/s$, 5834 $m^3/s$, and 5820 $m^3/s$, respectively. Compared to the predicted peak flow at *Zhengyangguan*, these scheduling plans achieved peak reductions of 56%, 53%, 51%, and 50%, respectively.

Although the basic genetic algorithm exhibits some randomness in generating results, directly comparing the merits of these four outcomes may not be entirely scientific. However, by comparing the peak reduction rates, it is evident that the effects of these four strategies are similar. This finding indicates that under current hydrological conditions, whether *MoziTan* and *Bailianya* Reservoirs participate in joint scheduling has a limited impact on the flow process at the *Zhengyangguan* section. In such cases, the scheduling strategy should focus more on effectively utilizing each reservoir's regulatory capabilities to achieve optimal flood control and water resource utilization efficiency.

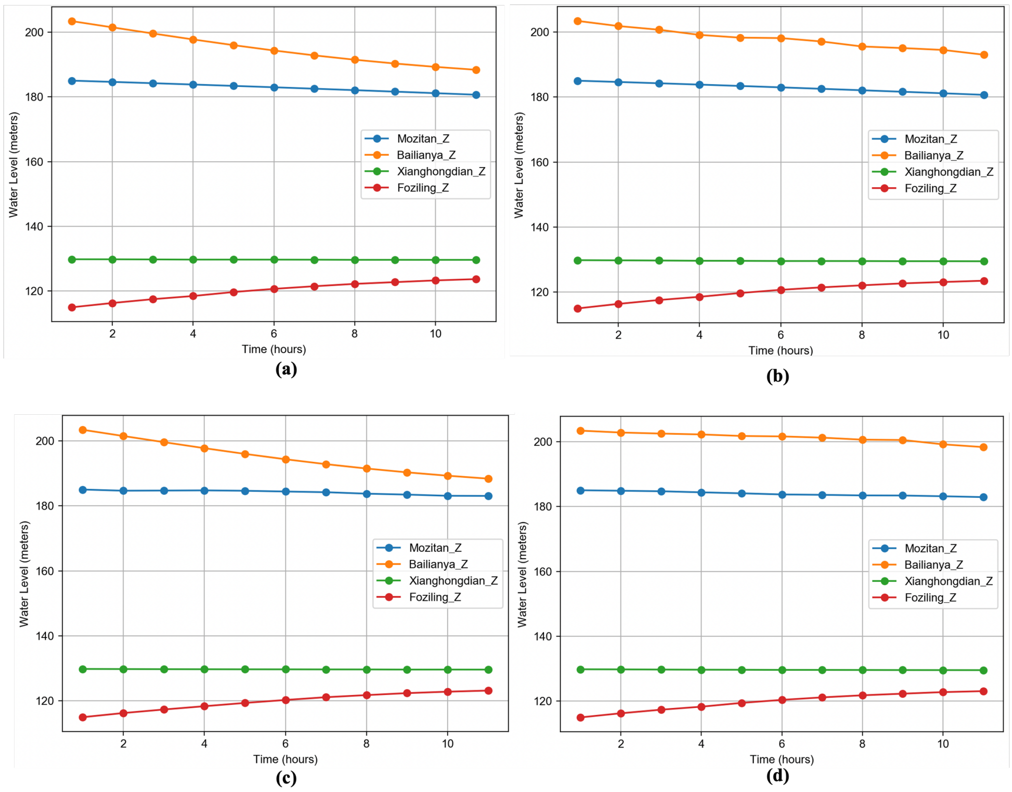


**Figure 11.** Water level changes corresponding to optimal outflow from reservoirs. (**a**) joint scheduling of Foziling and Xianghongdian Reservoirs; (**b**) joint scheduling of Xianghongdian, Foziling , and Bailianya Reservoirs; (**c**) joint scheduling of Xianghongdian, Foziling, and MoziTan Reservoirs; and (**d**) comprehensive joint scheduling of all the aforementioned reservoirs.

### *3.3. Multi-Objective Scheduling during Non-Flood Season*

As data on interval inflow and inflow to upstream reservoirs during the non-flood season were not available, the data from the flood season were proportionally reduced by a factor of 50 to simulate hydrological conditions during the non-flood season. Initial water levels were set at 160 m for *MoziTan* Reservoir, 190 m for *Bailianya* Reservoir, 122 m for *Xianghongdian* Reservoir, and 115 m for *Foziling* Reservoir. Additionally, the ecological flow for each reservoir was set at 50 cubic meters per second. Unlike the flood season, the scheduling cycle during the non-flood season is calculated daily, with each time step corresponding to the average daily flow.

After 100 generations of iteration, the algorithm generated the final generation of the population. A Pareto front was extracted from this population, with the level 0 Pareto front containing 97 individuals and the level 1 Pareto front containing 3 individuals. The distribution of the Pareto front is shown in Figure 12, where red indicates individuals on the level 0 Pareto front, and blue indicates individuals on the level 1 Pareto front. The horizontal axis represents the optimization objective $f_2$, where a higher value indicates a better individual; the vertical axis represents the optimization objective $f_3$, where a lower value indicates a better individual. Overall, individuals positioned in the top right corner excel in both objective functions.

When managing watershed resources, especially when facing critical challenges such as insufficient power generation or ecological flow, it is typical to select the most suitable optimization solution that meets current needs. Within the Pareto front, the optimal outflow processes for $f_2$ and $f_3$ are shown in Figure 13a,b, respectively.

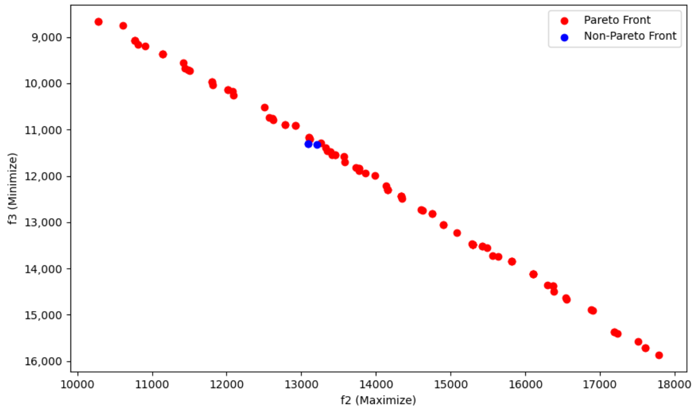


**Figure 12.** Pareto front.

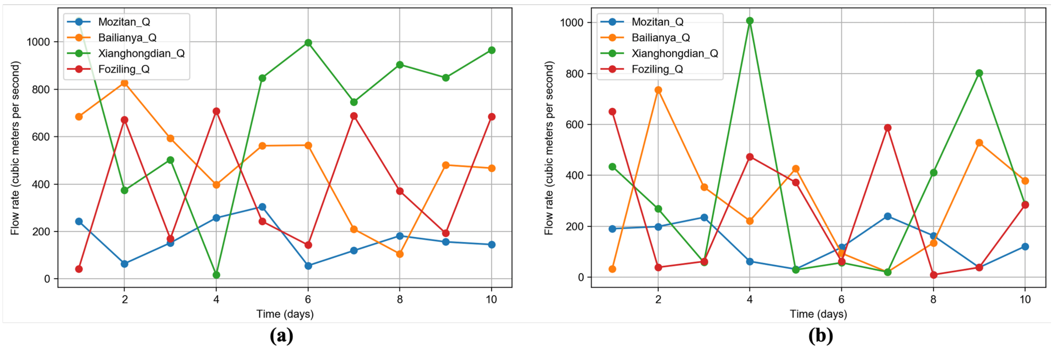


**Figure 13.** Best optimal outflow processes for $f_2$ and $f_3$.

*3.4. Generation of Scheduling Instructions*

This section conducted two experiments: first, converting the flow processes obtained by the basic genetic algorithm into scheduling actions, and second, converting the actual outflow of the reservoir into scheduling actions, using *Foziling* Reservoir as an example. *Foziling* Reservoir has three main flood discharge structures: a power generation channel, a flood discharge channel, and a spillway. The power generation channel must be opened, while the flood discharge channel can be opened or kept closed, with two open options (0 and 1). The spillway has six gates, each of which can be opened to degrees of 0, 1/6, 2/6, 3/6, 4/6, 5/6, or entirely (6/6), making seven possible degrees of openness [38]. Therefore, there are 14 possible combinations of openness settings for *Foziling* Reservoir. The water level–flow relationship for *Foziling* Reservoir is shown in Table 3.

As the water level in the reservoir changes over time, the discharge capacity of these 14 openness combinations will vary depending on the water level. Therefore, when generating scheduling instructions, it is necessary to choose the best combination of openness settings based on the current water level of the reservoir to achieve optimal discharge effects. The basic strategy is to calculate the maximum flow of the 14 openness combinations at the current water level and find the combination higher than and closest to the current outflow.

**Table 3.** The water level–flow relationship of Foziling Reservoir.

| Water Level (m) | Storage (billion $m^3$) | Discharge Building Release ($m^3$/s) | Power Generation | Flood Discharge Pipe | Spillway (6 Holes) | Total Discharge |
|---|---|---|---|---|---|---|
| 108.76 | 1.25 | 100 | 165 | 0 | 165 | 265 |
| 109.56 | 1.34 | 100 | 167 | 0 | 167 | 267 |
| 110.56 | 1.46 | 100 | 170 | 0 | 170 | 270 |
| 111.56 | 1.58 | 100 | 173 | 0 | 283 | 383 |
| 112.56 | 1.71 | 100 | 176 | 0 | 286 | 386 |
| 113.56 | 1.85 | 100 | 179 | 96 | 385 | 485 |
| 114.56 | 1.99 | 100 | 182 | 264 | 556 | 656 |
| 115.56 | 2.14 | 100 | 184 | 490 | 784 | 884 |
| 116.56 | 2.29 | 100 | 187 | 760 | 1057 | 1157 |
| 117.56 | 2.43 | 105 | 190 | 1075 | 1370 | 1470 |
| 118.56 | 2.62 | 105 | 193 | 1400 | 1698 | 1798 |
| 119.56 | 2.79 | 105 | 196 | 1780 | 2080 | 2180 |
| 120.56 | 2.97 | 100 | 199 | 2180 | 2479 | 2579 |
| 121.56 | 3.16 | 100 | 202 | 2600 | 2900 | 3000 |
| 122.56 | 3.35 | 100 | 205 | 3070 | 3375 | 3475 |
| 123.56 | 3.55 | 95 | 208 | 3575 | 3877 | 3977 |
| 124.56 | 3.75 | 95 | 210 | 4075 | 4380 | 4480 |
| 125.56 | 3.96 | 94 | 213 | 4644 | 4951 | 5051 |
| 126.56 | 4.18 | 0 | 216 | 5225 | 5441 | 5541 |
| 127.56 | 4.40 | 0 | 219 | 5850 | 6069 | 6169 |

The results of converting flow processes into scheduling actions for the joint scheduling scenario of *Xianghongdian* and *Foziling* using the basic genetic algorithm are shown in Table 4. In this table, the status column shows the degrees of openness of the power generation channel, flood discharge channel, and spillway at the current time. Moreover, the action column indicates the specific scheduling actions to be executed at that moment. The maximum flow column displays the maximum possible discharge flow the reservoir can achieve based on the current degrees of openness. The table shows that the errors between the maximum flows and the actual outflows are within 100 cubic meters per second, indicating that the conversion from flow processes to scheduling actions is relatively accurate.

**Table 4.** Conversion of Foziling Reservoir outflow from basic genetic algorithm to scheduling actions.

| Time | Water Level | Outflow | Status | Actions | Max Flow |
|---|---|---|---|---|---|
| 14:00 | 115 | 401 | (a, 5/6c) | a to 1, c to 5/6 | 402.8 |
| 15:00 | 116.3 | 173 | (b) | a to 0, b to 1, c to 0 | 186.22 |
| 16:00 | 117.5 | 255 | (a, 1/6c) | a to 1, b to 0, c to 1/6 | 280.7 |
| 17:00 | 118.4 | 64 | (a) | c to 0 | 105 |
| 18:00 | 119.7 | 385 | (a, 1/6c) | c to 1/6 | 410.3 |
| 19:00 | 120.6 | 436 | (a, 1/6c) | No change | 466.1 |
| 20:00 | 121.5 | 66 | (a) | c to 0 | 100 |
| 21:00 | 122.2 | 510 | (a, 1/6c) | c to 1/6 | 583.4 |
| 22:00 | 122.8 | 130 | (b) | a to 0, b to 1, c to 0 | 205.72 |
| 23:00 | 123.3 | 412 | (c) | b to 0, c to 1 | 522.95 |

Subsequently, the actual outflow process of the *Foziling* Reservoir was converted using data from the outflows between 18:00 on 1 July 2020, and 03:00 on 2 July 2020, as shown in Table 5. Through the conversion process, the maximum difference between the maximum flow and the actual outflow was 107 cubic meters per second, and the minimum difference was only two cubic meters per second. Such conversion results further validate the algorithm's effectiveness in scheduling for *Foziling* Reservoir.

**Table 5.** Conversion of actual monitored outflow from Foziling Reservoir to scheduling actions.

| Time | Water Level | Outflow | Status | Actions | Max Flow |
|---|---|---|---|---|---|
| 18:00 | 121.53 | 2480 | (6/6c) | a to 0, b to 0, c to 6/6 | 2587.4 |
| 19:00 | 121.3 | 2480 | (6/6c) | No change | 2490.8 |
| 20:00 | 121.08 | 2480 | (a, 6/6c) | a to 1 | 2498.4 |
| 21:00 | 120.88 | 2480 | (b, 6/6c) | a to 0, b to 1 | 2514.36 |
| 22:00 | 120.68 | 2480 | (a, b, 6/6c) | a to 1 | 2529.76 |
| 23:00 | 120.54 | 2020 | (a, b, 5/6c) | c to 5/6 | 2109.04 |
| 00:00 | 120.45 | 1690 | (a, b, 4/6c) | c to 4/6 | 1723.22 |
| 01:00 | 120.39 | 1671 | (a, b, 4/6c) | No change | 1707.34 |
| 02:00 | 120.33 | 1671 | (a, b, 4/6c) | No change | 1691.46 |
| 03:00 | 120.26 | 1671 | (a, b, 4/6c) | No change | 1672.9 |

## 4. Conclusions

This paper has developed an innovative genetic algorithm integrated with knowledge graph technology to optimize reservoir scheduling for diverse hydrological scenarios. The genetic algorithm, enhanced by the knowledge graph, facilitates the automatic construction and solving of optimization models, which adapts effectively across different watersheds. This integration addresses the need for custom model reconstruction for each unique watershed by encoding hydrodynamic simulations within genetic operations, thereby improving the practical utility of the outflow rates generated. Results from applying this methodology to the Huaihe basin demonstrated significant improvements in flood control during the flood season and optimized multi-objective scheduling for power generation and ecological preservation in the non-flood season.

Further exploration into integrating emerging technologies like real-time data processing and artificial intelligence could lead to more robust, predictive hydrological models. These advancements would ideally contribute to developing universally applicable, automated reservoir management systems that ensure efficient and effective water resource management across varying geographical and climatic conditions.

**Author Contributions:** Conceptualization, J.F. and S.Z.; methodology, H.T.; validation, H.T.; writing—review and editing, J.F., H.T. and S.Z. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by The National Key R&D Program of China (Grant No. 2023YFC3209203); The Water Conservancy Science and Technology Program of Jiangsu (Grant No. 2022002, 2023044); Major Science and Technology Program of The Ministry of Water Resources (Grant No. SKS-2022132); and The Fundamental Research Funds for the Central Universities (Grant No. B230201057).

**Data Availability Statement:** Data are contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Hanif, M.; Tahir, M.A.; Rafi, M. VRBagged-Net: Ensemble Based Deep Learning Model for Disaster Event Classification. *Electronics* **2021**, *10*, 1411. [CrossRef]
2. Bartos, M.; Kerkez, B. Pipedream: An interactive digital twin model for natural and urban drainage systems. *Environ. Model. Softw.* **2021**, *144*, 105120. [CrossRef]
3. Thaeer Hammid, A.; Awad, O.I.; Sulaiman, M.H.; Gunasekaran, S.S.; Mostafa, S.A.; Manoj Kumar, N.; Khalaf, B.A.; Al-Jawhar, Y.A.; Abdulhasan, R.A. A review of optimization algorithms in solving hydro generation scheduling problems. *Energies* **2020**, *13*, 2787. [CrossRef]

4. Kwon, J.-H.; Kim, M.-H.; Lee, J.-K.; Kim, H.-J.; Min, S.-G.; Lee, S. Development of a Heuristic Control Algorithm for Improving Energy Efficiency in Commercial Refrigerated Systems without Indoor Information. *Electronics* **2023**, *12*, 4678. [CrossRef]
5. Aksoy, A.; Valle, L.; Kar, G. Automated Network Incident Identification through Genetic Algorithm-Driven Feature Selection. *Electronics* **2024**, *13*, 293. [CrossRef]
6. Ren, M.; Zhang, Q.; Yang, Y.; Wang, G.; Xu, W.; Zhao, L. Research and application of reservoir flood control optimal operation based on improved genetic algorithm. *Water* **2022**, *14*, 1272. [CrossRef]
7. SeethaRam, K.V. Three level rule curve for optimum operation of a multipurpose reservoir using genetic algorithms. *Water Resour. Manag.* **2021**, *35*, 353–368. [CrossRef]
8. Ramírez, R.M.; Juárez, M.L.A.; Mora, R.D.; Morales, L.D.P.; Mariles, A.F.; Reséndiz, A.M.; Elizondo, E.C.; Paredes, R.B.C. Operation policies through dynamic programming and genetic algorithms, for a reservoir with irrigation and water supply uses. *Water Resour. Manag.* **2021**, *35*, 1573–1586. [CrossRef]
9. Shang, L.; Shang, Y.; Hu, L.; Li, J. Performance of genetic algorithms with different selection operators for solving short-term optimized reservoir scheduling problem. *Soft Comput.* **2020**, *24*, 6771–6785. [CrossRef]
10. Zhang, Z.; Zhang, S.; Wang, Y.; Jiang, Y.; Wang, H. Use of parallel deterministic dynamic programming and hierarchical adaptive genetic algorithm for reservoir operation optimization. *Comput. Ind. Eng.* **2013**, *65*, 310–321. [CrossRef]
11. Chang, J.; Bai, T.; Huang, Q.; Yang, D.-W. Optimization of water resources utilization by PSO-GA. *Water Resour. Manag.* **2013**, *27*, 3525–3540. [CrossRef]
12. Ahmadianfar, I.; Kheyrandish, A.; Jamei, M.; Gharabaghi, B. Optimizing operating rules for multi-reservoir hydropower generation systems: An adaptive hybrid differential evolution algorithm. *Renew. Energy* **2021**, *167*, 774–790. [CrossRef]
13. Moeini, R.; Soghrati, F. Optimum outflow determination of the multi-reservoir system using constrained improved artificial bee colony algorithm. *Soft Comput.* **2020**, *24*, 10739–10754. [CrossRef]
14. Hakimi-Asiabar, M.; Ghodsypour, S.H.; Kerachian, R. Deriving operating policies for multi-objective reservoir systems: Application of self-learning genetic algorithm. *Appl. Soft Comput.* **2010**, *10*, 1151–1163. [CrossRef]
15. Cheng, C.-T.; Wang, W.-C.; Xu, D.-M.; Chau, K.W. Optimizing hydropower reservoir operation using hybrid genetic algorithm and chaos. *Water Resour. Manag.* **2008**, *22*, 895–909. [CrossRef]
16. Wang, S.; Wang, Y.; Wang, Y.; Wang, Z. Comparison of multi-objective evolutionary algorithms applied to watershed management problem. *J. Environ. Manag.* **2022**, *324*, 116255. [CrossRef] [PubMed]
17. Sheng, D.; Lou, Y.; Sun, F.; Xie, J.; Yu, Y. Reengineering and its reliability: An analysis of water projects and watershed management under a digital twin scheme in China. *Water* **2023**, *15*, 3203. [CrossRef]
18. Li, X.; Cheng, G.; Lin, H.; Cai, X.; Fang, M.; Ge, Y.; Hu, X.; Chen, M.; Li, W. Watershed system model: The essentials to model complex human-nature system at the river basin scale. *J. Geophys. Res. Atmos.* **2018**, *123*, 3019–3034. [CrossRef]
19. Saab, S.M.; Othman, F.B.; Tan, C.G.; Allawi, M.F.; El-Shafie, A. Review on generating optimal operation for dam and reservoir water system: Simulation models and optimization algorithms. *Appl. Water Sci.* **2022**, *12*, 73. [CrossRef]
20. Zhang, R.; Zhang, S.; Wen, X.; Yue, Z.; Zhou, Y. Optimization of short-term hydropower scheduling with dynamic reservoir capacity based on improved genetic algorithm and parallel computing. *J. Hydrol.* **2024**, *636*, 131238. [CrossRef]
21. Gao, R.; Dong, Z.; Wang, Y.; Cui, Z.; Ye, M.; Dong, B.; Lu, Y.; Wang, X.; Song, Y.; Yan, S. Intelligent Cotton Pest and Disease Detection: Edge Computing Solutions with Transformer Technology and Knowledge Graphs. *Agriculture* **2024**, *14*, 247. [CrossRef]
22. He, L.; Ye, W.; Wang, Y.X.; Feng, H.S.; Chen, B.X.; Liang, D.Z. Using knowledge graph and RippleNet algorithms to fulfill smart recommendation of water use policies during shale resources development. *J. Hydrol.* **2023**, *617*, 128970. [CrossRef]
23. Feng, J.; Xu, X.; Lu, J. Construction and application of water conservancy information knowledge graph. *Comput. Mod.* **2019**, *9*, 35. [CrossRef]
24. Zeng, X.-L.; Zhang, G. Research and Application of Knowledge Map Based on the Integration and Sharing of Yellow River Resources. *People's Yellow River* **2021**, *43*, 282–284.
25. Zhang, Y.; Xiao, H.; Huang, J.-J.; Zhu, C.-F. Implementation of a Smart Water Conservancy System Based on Knowledge Graph and Microservices. *Inf. Technol. Water Resour.* **2021**, *4* , 17–21. [CrossRef]
26. Wang, X.; Zhou, J.; Ouyang, S.; Li, C. Research on joint impoundment dispatching model for cascade reservoir. *Water Resour. Manag.* **2014**, *28*, 5527–5542. [CrossRef]
27. Akbari, R.; Hessami-Kermani, M.-R.; Shojaee, S. Flood routing: Improving outflow using a new non-linear Muskingum model with four variable parameters coupled with PSO-GA algorithm. *Water Resour. Manag.* **2020**, *34*, 3291–3316. [CrossRef]
28. Zhang, Z.; Geng, S.; Jiang, X.; He, X.; Kan, G.; Yu, H.; Zhao, X. Study on Flood Control Optimal Operation of Reservoir Group-Taking *Pihe River Basin* as an Example. In Proceedings of the 2021 7th International Conference on Hydraulic and Civil Engineering & Smart Water Conservancy and Intelligent Disaster Reduction Forum (ICHCE & SWIDR), Nanjing, China, 6–8 November 2021; pp. 1613–1619. [CrossRef]
29. Feng, J.; Chang, Y.; Lu, J.; Tang, H.; Lv, H.; Qiu, Y. Construction and Application of Knowledge Graph for Water Engineering Scheduling Based on Large Language Model. *J. Front. Comput. Sci. Technol.* **2023**, *18*, 1637–1647. [CrossRef]
30. Marcelino, C.; Leite, G.; Delgado, C.; de Oliveira, L.; Wanner, E.; Jiménez-Fernández, S.; Salcedo-Sanz, S. An efficient multi-objective evolutionary approach for solving the operation of multi-reservoir system scheduling in hydro-power plants. *Expert Syst. Appl.* **2021**, *185*, 115638. [CrossRef]

31. Jia, D.; Zhang, T.; Wu, L.; Su, X.; Bai, T.; Huang, Q. Multi-objective cooperative optimization of reservoir scheduling and water resources allocation for inter-basin water transfer project based on multi-stage coupling model. *J. Hydrol.* **2024**, *630*, 130673. [CrossRef]
32. Li, Y.; Zhang, Z.; Kong, L.; Lei, X.; Zhu, J.; Li, H.; Wang, Y.; Cao, R. Hydraulic Optimization Control of Cascaded Open Channel under the Emergency Scenario of a Downstream Water Supply Interruption. *J. Water Resour. Plan. Manag.* **2023**, *149*, 04023029. [CrossRef]
33. Zhang, Z.; Geng, S.; He, X.; Zhang, S.; Tian, Y.; Li, H. Application of Improved Particle Swarm Optimization Algorithm in Reservoir Optimal Scheduling. In Proceedings of the 2nd International Conference on Artificial Intelligence, Big Data and Algorithms, CAIBDA 2022, Nanjing, China, 17–19 June 2022; VDE: Berlin, Germany, 2022; pp. 1–4.
34. Hu, W.-W.; Wang, G.-X.; Deng, W.; Li, S.-N. The influence of dams on ecohydrological conditions in the Huaihe River basin, China. *Ecol. Eng.* **2008**, *33*, 233–241. [CrossRef]
35. Yun, X.; Tang, Q.; Wang, J.; Liu, X.; Zhang, Y.; Lu, H.; Wang, Y.; Zhang, L.; Chen, D. Impacts of climate change and reservoir operation on streamflow and flood characteristics in the Lancang-Mekong River Basin. *J. Hydrol.* **2020**, *590*, 125472. [CrossRef]
36. Li, J.; Zhong, P.-A.; Wang, Y.; Liu, Y.; Zheng, J.; Yang, M.; Liu, W. Dynamic Self-Adaptive Modeling for Real-Time Flood Control Operation of Multi-Reservoir Systems. *Water* **2022**, *14*, 3740. [CrossRef]
37. Du, H.; Xia, J.; Zeng, S.; Tu, Y. Climate change and multi-reservoirs impacts on extreme flood: nonstationary regional frequency analysis and uncertainty estimation. *Water Resour. Manag.* **2024**, *38*, 951–965. [CrossRef]
38. Fan, Z.-W.; Zhang, M.; Su, Y.-L. Flood risk control of dams and dykes in middle reach of Huaihe River. *Water Sci. Eng.* **2014**, *7*, 17–31. [CrossRef]