

Article

A Scalarized Weighted-Sum Hybrid GA–PSO Decision-Support Framework for Constrained Water Resource Scheduling

Mehmet Akif Cifci ^{1,2,3} , Yousef Farhang ^{4,*}, Batuhan Öney ⁵ , Ziya Gökalp Ersan ⁶, Fazlı Yıldırım ⁷  and Uğur Akbulut ⁴ 

¹ Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam; meh.cifci@kvk.lt

² School of Engineering and Technology, Duy Tan University, Da Nang 550000, Vietnam

³ Department of Computer Engineering, Bandırma Onyedi Eylül University, Bandırma 10200, Turkey

⁴ Department of Computer Engineering, Faculty of Engineering and Architecture, Istanbul Esenyurt University, Istanbul 34510, Turkey

⁵ Department of Electronics and Automation, Vocational School, Istinye University, Istanbul 34396, Turkey

⁶ Department of Software Engineering, Faculty of Engineering and Architecture, Istanbul Gelisim University, Istanbul 34310, Turkey

⁷ Department of Management Information Systems, Faculty of Economics, Administrative and Social Sciences, Topkapı University, Istanbul 34087, Turkey

* Correspondence: youseffarhang@esenyurt.edu.tr

Abstract

Water resource management increasingly requires allocation methods that address rising demand, climate variability, and operational inefficiency. This study proposes an elite-transfer hybrid Genetic Algorithm–Particle Swarm Optimization framework for constrained water allocation under a weighted-sum scalarization model. The model combines four normalized objectives: minimizing water shortage, an operational cost coefficient, and allocation imbalance while maximizing utilization efficiency through an equivalent minimization term. A bidirectional elite-transfer mechanism links GA-based global exploration with PSO-based local refinement. The framework was evaluated using two capacity-constrained surrogate scenarios anchored to hydrometeorological records from Türkiye: Melekbahçe station (E21A033) in the Upper Euphrates Basin and Beşdeğirmen station (E12A003) in the Sakarya Basin. Daily streamflow records supported scenario construction, while precipitation and air-temperature data characterized local conditions. The proposed method was compared with standalone GA and standalone PSO, Differential Evolution, Grey Wolf Optimizer, a scalarized NSGA-II, adapted Kao–Zahara and Garg GA–PSO hybrids, and a no-elite ablation. All methods used the same objective formulation, normalization bounds, constraint-repair procedure, equal objective weights, tuning protocol, paired random seeds, and a budget of 10,000 objective-function evaluations. Performance was assessed through 30 paired runs per scenario. The proposed framework achieved the lowest mean scalar fitness values—0.1670 for Melekbahçe and 0.1752 for Beşdeğirmen—and reached the pre-defined convergence region after averages of 4587 and 4780 evaluations, respectively. All fitness and convergence improvements remained significant after Holm correction. Paired rank-biserial correlations ranged from 0.957 to 1.000 against the seven general comparators and were 0.824 and 0.781 against the no-elite ablation. The findings support the framework under the tested surrogate scenarios but do not establish Pareto-front dominance or universal superiority. Future work should examine measured operational data, additional basins, dynamic scheduling, alternative weights, and Pareto-based extensions.



Academic Editor: Mohd Shareduwan Mohd Kasihmuddin

Received: 3 July 2026

Revised: 31 July 2026

Accepted: 1 August 2026

Published: 3 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: hybrid optimization; water resource scheduling; genetic algorithm; particle swarm optimization; multi-objective optimization

1. Introduction

Water shortage has become one of the greatest challenges facing the world in the 21st century. The factors responsible for increased stress on fresh water supplies include population growth, urbanization, industrialization, and global warming [1]. Apart from the fact that they have caused water shortage, the aforementioned factors have made the supply of water difficult as well. The planning of water resources and efficient allocation of water have a very important role in managing the process of distribution, minimizing water losses, and ecological balance. However, conventional approaches in water resources management such as linear programming and rule-based water allocation do not consider the dynamic, nonlinear, and multiple objective nature of today's water resource systems. The methods are not adaptive and flexible to changing conditions [2,3].

In recent decades, artificial intelligence methods, particularly metaheuristic algorithms, have been widely applied to water resource optimization because of their ability to search large, nonlinear, and constrained solution spaces. Common approaches include the Genetic Algorithm (GA) [4,5], Particle Swarm Optimization (PSO) [6], Ant Colony Optimization, and Differential Evolution [2,7]. GA generally provides strong global exploration, whereas PSO offers rapid convergence and effective local refinement [7,8].

When applied independently, these algorithms may experience limitations such as premature convergence, reduced population diversity, or slow improvement during later search stages. Hybrid methods aim to reduce these limitations by combining complementary search mechanisms. GA–PSO approaches have therefore been used in various engineering optimization problems to balance exploration and exploitation [9–11]. Since GA–PSO hybridization is already well established, this study does not claim novelty from combining the two algorithms alone. Instead, its contribution lies in applying a bidirectional elite-transfer mechanism to a normalized and constrained water-allocation model within a reproducible weighted-sum scalarization framework. This work also extends the authors' previous studies on hybrid optimization and water-related modeling [12–15].

The study makes three main contributions. First, it formulates a constrained water resource scheduling model that integrates four normalized objective components: water shortage, an operational cost coefficient, allocation imbalance, and utilization efficiency. Second, it develops an elite-transfer GA–PSO framework in which high-quality GA solutions initialize the PSO refinement stage, while improved PSO solutions replace weaker individuals in the GA population. Third, it evaluates the proposed framework against standalone GA and PSO, Differential Evolution, Grey Wolf Optimizer, scalarized NSGA-II, two adapted published GA–PSO methods, and a no-elite ablation. All methods use the same objective formulation, normalization bounds, constraint-repair procedure, parameter tuning protocol, paired random seeds, and objective-function evaluation budget.

The experimental design includes 30 paired runs for each scenario, Wilcoxon signed-rank tests with Holm correction, paired rank-biserial effect sizes, bootstrap confidence intervals, and detailed documentation of preprocessing, scenario construction, parameter tuning, constraint handling, and evaluation accounting. These procedures support the reliability and reproducibility of the comparison but are not presented as independent scientific contributions.

The study adopts a computer science and optimization methodology perspective and should be interpreted as a proof of concept based on surrogate allocation scenarios an-

chored to real hydrometeorological records. The model does not use measured operational demand, allocation, capacity, or accounting-based cost records from water authorities. Practical deployment would therefore require validation with operational datasets and assessment by water resource engineers, practitioners, and relevant authorities. The conclusions are limited to the selected scenarios, scalarization strategy, parameter settings, and experimental protocol.

2. Related Work

Conventional techniques such as Linear Programming (LP), Nonlinear Programming (NLP), and Dynamic Programming (DP) have long been applied to reservoir operation, water allocation, and irrigation scheduling, but they scale poorly to nonlinear, multi-objective, large-scale water systems, and DP in particular suffers from the curse of dimensionality [16]. Metaheuristics address these weaknesses: the Genetic Algorithm (GA) offers strong global search [4] and Particle Swarm Optimization (PSO) offers rapid convergence and simplicity [6,8], and Ant Colony Optimization and Differential Evolution have also been applied to nonlinear water resource problems [17]. Each method has known drawbacks, including slow late-stage convergence for GA and premature convergence for PSO [18], which motivates hybridization. Recent water resource studies continue this line with hybrid metaheuristics for reservoir operation and allocation [19].

To address these weaknesses, hybrid approaches combine complementary search mechanisms, and GA–PSO hybridization in particular outperforms each method individually [20]. Hybrid GA–PSO designs differ mainly in how the two search processes exchange information. The researchers combined GA and PSO in a scheme in which the population is split and the two operators evolve and re-rank a shared pool of candidates [9]. The researcher enhanced the upper half of each generation with PSO before applying GA recombination to evolve recurrent networks [10]. The researcher alternated PSO-driven search with GA operators for constrained engineering design problems [21]. Related variants couple PSO or GA with clustering or with operator-level mutation customization [22], and recent work extends this line with enhanced PSO–GA hybrids for large-scale service placement [23]. Compared with these designs, the framework proposed here uses a bidirectional elite-transfer rule tailored to constrained water allocation: elite GA individuals seed the PSO refinement phase, improved PSO particles re-enter the GA population by replacing its weakest members, and every transferred solution passes through the same feasibility repair procedure. The contribution is therefore not the hybridization of GA and PSO as such but the combination of this elite exchange with a normalized, constraint-repaired, and reproducible scalarized water scheduling formulation. Multi-objective metaheuristics have also been applied directly to water systems. The researchers used multi-objective PSO to generate trade-offs in reservoir operation [16], and authors applied multi-objective PSO to water distribution system design [17]. The researchers introduced NSGA-II [18], which remains a standard Pareto-based baseline in this area within the broader development of evolutionary multi-objective optimization [20]. In parallel, machine learning models such as ANN, SVM, and LSTM networks forecast water demand, rainfall, and reservoir inflow as inputs to optimization [24], and Internet of Things (IoT) infrastructures for smart water management couple such forecasting and optimization with sensor networks and predictive analytics [25].

In conclusion, the reviewed literature shows that hybrid metaheuristic methods offer a practical approach to complex water resource scheduling problems by combining complementary search capabilities. However, published GA–PSO variants differ considerably in their information-exchange mechanisms, and direct comparisons under the same constrained, normalized, and repair-based water-allocation formulation remain limited.

To address this gap, the present study evaluates the proposed bidirectional elite-transfer framework against standalone GA, standalone PSO, DE, GWO, scalarized NSGA-II, the adapted Kao–Zahara and Garg hybrid GA–PSO methods, and a no-elite ablation. All methods are assessed using the same decision representation, scalar objective, normalization bounds, constraint-repair procedure, parameter tuning protocol, paired random seeds, and objective-function evaluation budget. This design allows the contribution of the elite-transfer mechanism to be examined separately from the general effect of combining GA and PSO.

3. Problem Formulation

To formulate the water resource scheduling problem, one needs to define it as a multi-objective optimization problem, where multiple objectives must be optimized at once within certain constraints. The objective is to optimize the distribution of water resources among different geographical regions. Due to the natural constraints of the problem in terms of cost, equity, and performance, the problem is formulated as a multi-objective optimization problem. In this study, the managed system is defined at the river-basin scale: each station-based scenario represents the allocation of water made available by a gauged stream to a set of competing allocation units, subject to availability, demand, non-negativity, and capacity constraints. Reservoir routing, irrigation canal hydraulics, and groundwater wells are not modeled explicitly.

3.1. Objectives

The water resource scheduling problem is formulated as a constrained scalarized multi-objective optimization problem. Let N denote the number of allocation units. For each allocation unit i , D_i denotes water demand, W_i denotes the allocated water, C_i denotes the operational cost coefficient, T denotes the total available water, and $W_i^{\max}$ denotes the maximum delivery capacity. The optimization model considers four objective components: water shortage, operational cost, allocation imbalance, and utilization efficiency.

(a) Minimization of Water Shortage

Water shortage occurs when the allocated water is lower than the required demand. The objective is to minimize the total unmet demand across all allocation units. The normalized water-shortage objective is defined as

$$F_1 = \frac{\sum_{i=1}^N \max(0, D_i - W_i)}{\sum_{i=1}^N D_i} \quad (1)$$

where D_i is the water demand of allocation unit i , W_i is the allocated water of allocation unit i , and N is the total number of allocation units. A lower F_1 value indicates lower unmet demand and therefore better shortage reduction.

(b) Minimization of Operational Cost

The operational cost objective is calculated as the total cost-related allocation value across all allocation units:

$$F_2 = \sum_{i=1}^N C_i W_i \quad (2)$$

where C_i is the unit operational cost coefficient for supplying water to allocation unit i . In this study, C_i is treated as a normalized model coefficient representing relative extraction, treatment, conveyance, or allocation difficulty. It should not be interpreted as a directly measured monetary cost unless direct accounting-based cost data are available.

(c) Minimization of Allocation Imbalance

To evaluate fairness among allocation units with different demand levels, the allocation satisfaction ratio is first defined as

$$S_i = \frac{W_i}{D_i} \quad (3)$$

The average satisfaction ratio is then calculated as

$$\bar{S} = \frac{1}{N} \sum_{i=1}^N S_i \quad (4)$$

The allocation-imbalance objective is defined as the variance of the satisfaction ratios:

$$F_3 = \frac{1}{N} \sum_{i=1}^N (S_i - \bar{S})^2 \quad (5)$$

where S_i represents the satisfaction ratio of allocation unit i and $\bar{S}$ represents the average satisfaction ratio across all allocation units. A lower F_3 value indicates a more balanced allocation. This ratio-based formulation is preferred because it evaluates fairness relative to demand satisfaction rather than only using absolute water volumes.

(d) Maximization of Utilization Efficiency

Utilization efficiency is defined as the ratio of effectively allocated water to the total available water:

$$F_4 = \frac{\sum_{i=1}^N \min(W_i, D_i)}{T} \quad (6)$$

where T denotes the total available water. Since the optimization framework is formulated as a minimization problem, the utilization objective is transformed into a minimization form as follows:

$$F'_4 = 1 - F_4 \quad (7)$$

A lower F'_4 value indicates higher utilization efficiency.

3.2. Combined Multi-Objective Function

The four objective components are combined using a weighted-sum scalarization approach. Before aggregation, each objective component is normalized to the interval $[0, 1]$ so that objectives with larger numerical scales do not dominate the scalar fitness function. The final scalar fitness function is defined as

$$Fitness = w_1 f_1 + w_2 f_2 + w_3 f_3 + w_4 f_4 \quad (8)$$

where *Fitness* is the overall scalar objective function. The terms f_1 , f_2 , f_3 , and f_4 represent the normalized objective components associated with water shortage, the operational cost coefficient, allocation imbalance, and transformed utilization inefficiency, respectively. The terms w_1 , w_2 , w_3 , and w_4 are non-negative weight coefficients that determine the relative importance of each objective. The weight coefficients satisfy the following conditions:

$$w_1 + w_2 + w_3 + w_4 = 1 \quad (9)$$

$$w_j \geq 0, j = 1, 2, 3, 4 \quad (10)$$

A higher w_1 gives more importance to reducing the water shortage. A higher w_2 gives more importance to reducing the operational cost coefficient. A higher w_3 gives more importance to balanced allocation. A higher w_4 gives more importance to improving utilization efficiency.

The weighted-sum method is a practical way to combine multiple objectives into a single scalar optimization problem. However, the obtained solution depends on the selected weight values. Therefore, the results reported in this study should be interpreted as scalarized multi-objective optimization results under the selected weight configuration. This formulation does not generate the complete Pareto front, especially when the optimization landscape is nonlinear or non-convex. For this reason, Pareto-based approaches such as NSGA-II, MOEA/D, or multi-objective PSO may complement the proposed scalarized framework in future studies.

3.3. Constraints

The proposed optimization model is subject to physical and operational constraints to ensure feasible water allocation.

i. Water Availability Constraint. The total allocated water must not exceed the total available water:

$$\sum_{i=1}^N W_i \leq T \quad (11)$$

where T denotes the total available water.

ii. Non-negativity Constraint. The allocated water for each allocation unit must be non-negative:

$$W_i \geq 0, \forall i \quad (12)$$

This constraint ensures that negative water allocation values cannot occur.

iii. Demand Satisfaction Constraint. The allocated water for each allocation unit cannot exceed its demand:

$$W_i \leq D_i, \forall i \quad (13)$$

where D_i denotes the water demand of allocation unit i . This constraint prevents over-allocation beyond actual demand.

iv. System Capacity Constraint. Each allocation unit or delivery channel may have a maximum delivery capacity:

$$W_i \leq W_i^{\max}, \forall i \quad (14)$$

where $W_i^{\max}$ denotes the maximum delivery capacity of allocation unit i .

3.4. Remarks

The formulation above is nonlinear, multi-constraint, and multidimensional: demands, availability, cost, fairness, and efficiency interact, and deterministic approaches such as linear programming, dynamic programming, and enumeration become ineffective as scale and uncertainty grow [26]. A second difficulty is goal conflict. In this model, decision-makers pursue four main objectives: minimizing the operational cost coefficient, minimizing water shortage, maximizing allocation fairness, and maximizing utilization efficiency. These objectives conflict to some degree. For instance, reducing operating costs leads to unfair water distribution, whereas maximizing allocation fairness leads to higher costs. These properties motivate the metaheuristic approach adopted here: GA and PSO offer complementary global exploration and fast exploitation behavior with known individual weaknesses [4,6,7], and their hybridization, detailed in Section 4, targets a balance of exploration and exploitation for exactly this problem class.

The weighted-sum scalarization used in this research reduces the multi-objective problem to a single-objective one that metaheuristic methods can optimize directly, and the weight coefficients allow decision-makers to encode policy or operational priorities. Three limitations follow from this choice: the approach cannot recover all Pareto-optimal solutions when the landscape is nonlinear or non-convex, the obtained solution depends

on the selected weights, and explicit trade-off analysis is not available within a single optimization run. The results of this study are therefore weight-dependent scalarized outcomes, and Pareto-based methods such as NSGA-II [18], MOEA/D, or multi-objective PSO [16,17] remain complementary directions. The experiments are based on a real hydrometeorological dataset, and the findings should be read within the limits of the selected data, static scheduling formulation, normalization procedure, and weight configuration. Extensions to additional regions, longer horizons, and operational datasets are identified in Section 7.

4. Materials and Methods

This section specifies the proposed elite-transfer hybrid GA–PSO framework and the experimental methodology. Each candidate solution is a feasible or repaired water-allocation vector evaluated with the scalar fitness function of Section 3 under the availability, non-negativity, demand, and capacity constraints. The GA phase provides global exploration, and the PSO phase provides local refinement, as detailed in Sections 4.1–4.5.

4.1. Genetic Algorithm Component

The Genetic Algorithm provides the global exploration component of the proposed framework. Each individual in the GA population represents a candidate water-allocation vector:

$$X = (W_1, W_2, \dots, W_N)$$

where W_i denotes the allocated water for allocation unit i and N denotes the number of allocation units.

The main GA operations used in this study are as follows:

- ⇒ Initialization: candidate allocation vectors are generated within the feasible allocation limits.
- ⇒ Fitness evaluation: each individual is evaluated using the scalarized objective function defined in Section 3.
- ⇒ Selection: tournament selection is used to choose parent solutions with better fitness values.
- ⇒ Crossover: selected parent solutions exchange allocation information to generate offspring.
- ⇒ Mutation: small random changes are introduced to maintain population diversity and reduce premature convergence.
- ⇒ Constraint handling: infeasible allocation values are repaired according to the water availability, demand, non-negativity, and capacity constraints.

Because GA can converge slowly in later iterations, the framework transfers high-quality GA individuals to the PSO component for local refinement (Section 4.4).

4.2. Particle Swarm Optimization Component

Particle Swarm Optimization provides the local exploitation component. Particles are not initialized randomly in every cycle; elite individuals selected from the GA population seed the swarm, so the local search begins from promising regions identified by GA.

The velocity update equation is defined as

$$v_i^{t+1} = wv_i^t + c_1r_1(pbest_i^t - x_i^t) + c_2r_2(gbest^t - x_i^t) \quad (15)$$

The position update equation is defined as

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (16)$$

where x_i^t is the position of particle i at iteration t , v_i^t is the velocity of particle i , $pbest_i^t$ is the personal-best position of particle i , $gbest^t$ is the best position found by the swarm, w is the inertia weight, c_1 and c_2 are acceleration coefficients, and r_1 and r_2 are random values in the interval $[0, 1]$. After each position update, the candidate allocation vector is checked against the constraints. If a particle violates the feasibility conditions, its position is repaired before the fitness value is recalculated. This prevents invalid water-allocation vectors from being used in later optimization steps.

4.3. Constraint Handling and Repair Procedure

All algorithms in this study apply the same deterministic two-step repair procedure whenever a candidate allocation vector violates the constraints of Section 3.3. In the first step, each allocation component is clipped to its box bounds: it is raised to zero if it is negative and lowered to the smaller of the corresponding demand value and the maximum delivery capacity if it exceeds that limit. This step enforces the non-negativity, demand satisfaction, and capacity constraints in Equations (12)–(14). In the second step, the water availability constraint in Equation (11) is checked. If the total clipped allocation does not exceed the total available water, the vector is accepted without further modification. Otherwise, every component is multiplied by the ratio of the total available water to the total clipped allocation. This proportional rescaling restores feasibility with respect to Equation (11) while preserving the relative allocation pattern, and because the scaling factor is smaller than one, the rescaled components remain within their box bounds, so no additional repair pass is required. The repair operator is applied at the same points in every algorithm: after population or swarm initialization, after crossover and mutation in the GA phase, and after each velocity and position update in the PSO phase. No penalty terms are added to the fitness function, so all reported fitness values are computed on feasible vectors only. Because the identical repair rule is shared by GA, PSO, DE, GWO, the scalarized NSGA-II baseline, and both hybrid variants of the ablation study, performance differences among the compared methods cannot be attributed to differences in constraint handling.

4.4. Elite-Transfer Hybrid GA–PSO Framework

The proposed Hybrid GA–PSO framework uses an elite-transfer mechanism between the GA and PSO phases. In each hybrid cycle, the GA population is first evolved through selection, crossover, and mutation. After evaluating the updated GA population, the best-performing individuals are selected as the elite set. These elite solutions are then transferred to the PSO phase as initial particles.

Figure 1 illustrates the complete workflow of the proposed framework, including GA-based exploration with constraint repair, elite selection and transfer, PSO-based refinement with constraint repair, replacement of weak GA individuals, and the termination check. When the termination condition is not satisfied, the branch labeled “No” returns the updated population to the scalar fitness evaluation stage, thereby closing the optimization loop. Once the termination condition is met, the algorithm returns the best feasible water-allocation solution.

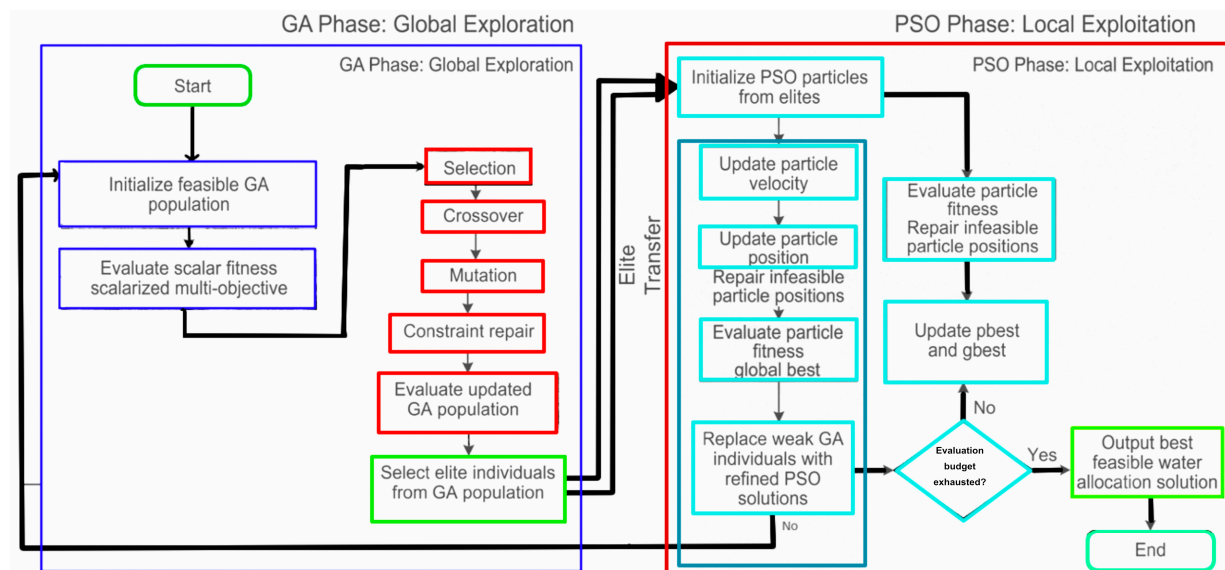


Figure 1. Structure of the proposed elite-transfer hybrid GA–PSO optimization framework for scalarized water resource scheduling. The workflow combines GA-based global exploration with PSO-based local refinement through elite solution transfer. When the termination criterion is not satisfied, the No branch returns the updated population to the scalar fitness evaluation step and a new hybrid cycle begins.

During the PSO refinement phase, the transferred elite particles are updated using the velocity and position equations and then evaluated with the same scalar fitness function. When the refined PSO solutions improve the overall population quality, they replace the worst-performing individuals in the GA population. This replacement strategy preserves useful genetic diversity while reinserting locally refined solutions into the population for the next optimization cycle.

4.5. Algorithmic Procedure

The complete procedure of the proposed elite-transfer hybrid GA–PSO algorithm is summarized in Algorithm 1.

Each complete hybrid cycle uses 480 objective-function evaluations: 40 evaluations for the GA offspring, 40 evaluations for the newly initialized non-elite PSO particles, and 400 evaluations for eight PSO updates involving 50 particles. Together with the 50 evaluations used for the initial GA population, 20 complete cycles consume 9650 evaluations. The remaining 350 evaluations are assigned to seven additional updates of the existing PSO swarm, ensuring that the algorithm uses exactly the same 10,000-evaluation budget as the comparison methods.

The algorithm combines GA-based global exploration with PSO-based local refinement through a bidirectional elite-transfer mechanism. High-quality GA candidates guide the PSO search toward promising areas, while improved PSO solutions replace weaker GA individuals. The common repair operator ensures that every evaluated allocation vector satisfies the non-negativity, demand, capacity, and total-availability constraints. The use of a global objective-function evaluation counter also ensures that computational effort is controlled consistently across the proposed method and the competing algorithms.

Algorithm 1. Elite-transfer hybrid GA–PSO algorithm for scalarized water resource scheduling.

Input:

GA population size $N_{GA} = 50$, PSO swarm size $N_{PSO} = 50$, maximum objective-function evaluation budget $E_{max} = 10,000$, number of GA offspring per cycle $N_{off} = 40$, crossover rate, mutation rate, inertia weight, acceleration coefficients, elite-transfer ratio $\rho = 0.20$, number of PSO refinement updates per complete cycle $R = 8$, objective weights, normalization bounds, and constraint parameters.

Output:

Best feasible water-allocation vector and its scalar fitness value.

Step 1. Set the objective-function evaluation counter to $E = 0$.

Step 2. Initialize a GA population containing 50 feasible candidate allocation vectors.

Step 3. Evaluate all GA individuals using the scalar fitness function, and update the evaluation counter: $E \leftarrow E + 50$.

Step 4. Store the best feasible solution as the current global incumbent.

Step 5. While sufficient budget remains for a complete hybrid cycle, perform the following steps:

Step 5.1. Select parent individuals from the GA population using tournament selection.

Step 5.2. Apply crossover and mutation to generate 40 offspring solutions.

Step 5.3. Repair each infeasible offspring using the deterministic constraint-handling procedure described in Section 4.3.

Step 5.4. Evaluate the 40 repaired offspring, and update the evaluation counter:

$E \leftarrow E + 40$.

Step 5.5. Combine the parent and offspring populations, and retain the 50 best feasible individuals as the updated GA population.

Step 5.6. Select the 10 best GA individuals, an elite-transfer ratio of 20%, as the elite set.

Step 5.7. Initialize a 50-particle PSO swarm using the 10 elite GA individuals and 40 newly generated feasible particles. Reuse the existing fitness values of the elite particles, and evaluate only the 40 newly generated particles: $E \leftarrow E + 40$.

Step 5.8. Initialize the personal-best position of each particle and determine global best.

Step 5.9. Perform eight PSO refinement updates. During each update:

a. Update particle velocities using Equation (15).

b. Update particle positions using Equation (16).

c. Repair infeasible particle positions to the procedure in Section 4.3.

d. Evaluate all 50 updated particles.

e. Update the evaluation counter: $E \leftarrow E + 50$.

f. Update the personal-best and global-best positions.

Step 5.10. Rank the PSO-refined particles according to scalar fitness.

Step 5.11. Replace the weakest GA individuals with PSO-refined particles only when the transferred solutions provide lower scalar fitness values.

Step 5.12. Update the global incumbent when a better feasible solution is identified.

Step 6. After completing 20 full hybrid cycles, the evaluation counter reaches 9650. Use the remaining 350 evaluations for seven additional PSO refinement updates of the existing 50-particle swarm. During each additional update, repeat the velocity update, position update, constraint repair, fitness evaluation, and personal-best and global-best updates described in Step 5.9.

Step 7. Stop when the evaluation counter reaches exactly $E = E_{max} = 10,000$.

Step 8. Return the best feasible water-allocation vector and its final scalar fitness value.

4.6. Advantages and Limitations of the Proposed Hybrid Model

The hybrid design offers four methodological advantages: it combines the diversity-preserving ability of GA with the fast convergence behavior of PSO; the elite-transfer mechanism lets the PSO phase start from promising candidates rather than random positions; the scalarized formulation optimizes the four normalized objective components of Section 3 within one weighted fitness function; and the framework adapts to different scheduling scenarios through the objective weights and constraint parameters. Its limitations mirror the design choices: results depend on the weight configuration, normalization, dataset, and parameter settings; weighted-sum scalarization does not generate a Pareto front; and the experiments use real hydrometeorological records under a static scheduling formulation, so validation with operational allocation records, measured cost data, dynamic demands, and longer horizons remains necessary.

4.7. NSGA-II Comparison Strategy

NSGA-II is a Pareto-based multi-objective optimization algorithm, whereas the proposed hybrid GA–PSO framework optimizes a scalar weighted-sum objective. Therefore, the comparison with NSGA-II requires careful interpretation. In this study, NSGA-II generated a population of candidate allocation solutions, and the solution with the lowest value under the common normalized weighted-sum function was selected for scalar comparison with the other algorithms.

This comparison does not indicate that the proposed framework dominates NSGA-II in terms of Pareto-front quality. It evaluates only whether the proposed hybrid GA–PSO obtains a lower scalar fitness value under the same objective weights, normalization bounds, constraint-repair rules, scenario definitions, and objective-function evaluation budget. A complete Pareto-based comparison would require indicators such as hypervolume, spread, spacing, and inverted generational distance. Such an analysis is outside the scalarized decision-support scope of the present study.

4.8. Dataset and Scenario Construction

Hydrometeorological records previously analyzed by Akbulut et al. [13] were used to describe the environmental contexts of the two station-based cases. The optimization demand, availability, capacity, and delivery-burden parameters were constructed separately and were not directly observed operational quantities. The dataset contains daily average streamflow, total daily precipitation, and daily average air-temperature observations from two hydrological regions in Türkiye.

The first study area is the Melekbahçe stream-gauging station (E21A033) in the Munzur sub-basin of the Upper Euphrates Basin; the second is the Beşdeğirmen stream-gauging station (E12A003) in the Porsuk Creek sub-basin of the Sakarya Basin [27]. The corresponding precipitation and air-temperature observations were obtained from the Tunceli (17,165) and Kütahya (17,155) meteorological stations, respectively. Only overlapping periods containing all required observations were used: 1997–2022 for Melekbahçe and 1980–2022 for Beşdeğirmen; Kütahya meteorological observations after 2022 were excluded because corresponding streamflow observations were unavailable, which prevented temporal mismatch among the variables. Table 1 lists the observation periods and all scenario parameters.

Daily average streamflow provided the hydrological basis for characterizing the supply conditions of the surrogate allocation scenarios. Precipitation and air temperature were used to describe the hydrometeorological conditions of the two study areas but did not enter the four optimization objectives directly. Because the source dataset does not contain accounting-based water-distribution costs, the operational cost coefficients were represented by normalized dimensionless parameters reflecting relative extraction,

treatment, conveyance, or supply difficulty. These coefficients should not be interpreted as measured monetary costs.

Table 1. Scenario construction parameters for the Melekbahçe and Beşdeğirmen station-based allocation scenarios.

Parameter	Melekbahçe (E21A033), Upper Euphrates Basin	Beşdeğirmen (E12A003), Sakarya Basin
Number of allocation units, N	8	10
Hydrometeorological record period	1997–2022, 26 years	1980–2022, 43 years
Scenario availability, T	1850	3450
Demand vector, D_i	[320, 280, 250, 220, 180, 150, 120, 100]	[450, 380, 320, 290, 250, 210, 180, 150, 120, 90]
Total demand, $\sum_i D_i$	1620	2440
Cost-coefficient vector, C_i	[0.65, 0.55, 0.75, 0.45, 0.80, 0.60, 0.40, 0.70]	[0.70, 0.50, 0.80, 0.60, 0.45, 0.75, 0.55, 0.65, 0.40, 0.85]
Capacity vector, $W_i^{\max}$	[280, 240, 210, 190, 160, 130, 100, 85]	[380, 320, 270, 240, 210, 180, 150, 130, 100, 75]
Total capacity, $\sum_i W_i^{\max}$	1395	2055
Demand-to-supply ratio, $\sum_i D_i / T$	0.876	0.707
Capacity-to-demand ratio, $\sum_i W_i^{\max} / \sum_i D_i$	0.861	0.842
Availability construction	Scenario-assumed planning value informed by the relative streamflow characteristics of the study area	Scenario-assumed planning value informed by the relative streamflow characteristics of the study area
Allocation-unit interpretation	Hypothetical competing irrigation, domestic, industrial, and environmental demand units	Same interpretation
Parameter status	Observed hydrometeorological records, derived streamflow information, and scenario-assumed allocation parameters	Same classification

Note: T , D_i , and $W_i^{\max}$ are expressed in units of 10^3 m^3 . The availability, demand, capacity, and operational cost-coefficient values are scenario parameters informed by the hydrological characteristics of the study areas and should not be interpreted as directly measured operational quantities.

The hydrometeorological records were preprocessed before scenario construction. Missing observations were treated according to the location and duration of each missing segment. Linear interpolation was used for short internal gaps, backward filling was used for boundary gaps when a preceding observation was unavailable, and mean imputation was applied when interpolation was not appropriate. Potential outliers were screened using Z-score analysis, Tukey’s fences, and interquartile-range analysis. Flagged observations were examined with respect to the expected physical behavior of the corresponding variable and were modified or excluded only when they were considered inconsistent with the hydrological record. The same preprocessing procedure was applied before constructing the inputs used by all optimization algorithms. Preprocessing was not performed separately for individual algorithms and therefore could not introduce algorithm-specific differences.

Long-memory characteristics, including the Hurst coefficient, were not included because the present study addresses static allocation optimization using constructed scenario parameters rather than sequential streamflow forecasting. Persistence-aware scenario gen-

eration and dynamic allocation based on long-memory diagnostics remain directions for future research.

Three information levels were distinguished during scenario construction. The first consisted of the observed daily streamflow, precipitation, and air-temperature records. The second consisted of daily water volumes derived from the streamflow observations. For a daily average discharge Q_t , expressed in m^3/s , the corresponding daily volume was calculated as $V_t = Q_t \times 86,400$, where 86,400 is the number of seconds in one day. The resulting volume V_t was expressed in m^3/day and converted to units of 10^3 m^3 when used to characterize the hydrological scale of the scenarios.

The third information level consisted of the static allocation-scenario parameters, including the number of allocation units N , total available water T , demand vector D_i , operational cost-coefficient vector C_i , and delivery-capacity vector $W_i^{\max}$. The multi-year observation periods were used to characterize the hydrological conditions of the two study areas and were not treated as multi-year optimization horizons.

The scenario-availability values were constructed planning assumptions informed by the relative streamflow characteristics of the two study areas. They were not obtained by directly summing the complete daily streamflow records and should not be interpreted as measured operational availability totals. The values $T = 1850$ for Melekbağçe and $T = 3450$ for Beşdeğirmen were used as rounded scenario specifications representing the relative supply scale of the two cases.

The demand values were constructed as $D_i = k_i B$, where $B = 10$, expressed in units of 10^3 m^3 , and k_i denotes the unit-specific demand factor. The demand factors are scenario assumptions rather than measured sectoral demands. The hypothetical allocation units represent competing irrigation, domestic, industrial, and environmental water uses within the service area of each station. The allocation bounds ensure that each assigned volume remains non-negative, does not exceed the corresponding demand, and remains within the unit-level delivery capacity. Precipitation and air temperature do not directly enter the objective components F_1 to F_4 . They provide contextual information about the hydrometeorological conditions of the study areas and support interpretation of the streamflow records. The operational cost coefficients C_i are fixed dimensionless parameters in the interval $[0, 1]$. Larger values represent greater assumed extraction, treatment, conveyance, or supply difficulty. The same coefficient vector was applied to every algorithm and replication block within each scenario.

The demand factors, operational cost coefficients, delivery capacities, and scenario-availability values were not calibrated using measured operational records from a water authority. The model should therefore be interpreted as a basin-scale allocation-planning proof of concept rather than a real-time reservoir, canal, or distribution-system operating model. The sensitivity analyses in Section 5.5 examine the influence of changes in these assumptions but do not replace calibration using measured operational demand, capacity, availability, and cost data.

All algorithms used the same preprocessed records, scenario definitions, and paired random seeds; the common evaluation protocol, including the strict budget of 10,000 objective-function evaluations per run, is specified in Sections 4.9 and 4.10.

Because the objective components have different numerical scales, each component was normalized to the interval $[0, 1]$ before scalar aggregation. Min–max normalization was defined as

$$F_{\text{norm}} = \frac{F - F_{\min}}{F_{\max} - F_{\min}}, \quad (17)$$

where F is the original objective value, $F_{\min}$ and $F_{\max}$ are the common lower and upper reference bounds for the corresponding objective component, and F_{norm} is the normalized value. Within each scenario, the same normalization bounds were fixed before final testing

and applied to every algorithm and replication block. No algorithm-specific or run-specific rescaling was performed. This procedure ensured direct comparability of the resulting objective-component and scalar fitness values.

The normalized objective components were combined using the weighted-sum function defined in Equation (8). The terms f_1 , f_2 , f_3 , and f_4 denote the four normalized objective components defined in Section 3. The balanced weight configuration $w = (0.25, 0.25, 0.25, 0.25)$ was used in the primary comparative experiments. Alternative weight configurations were examined separately in Section 5.5.

The two cases should therefore be interpreted as capacity-constrained surrogate allocation scenarios anchored to real hydrometeorological records. Table 1 distinguishes the observed hydrometeorological records and derived streamflow information from the scenario-assumed availability, demand, capacity, and operational cost-coefficient parameters used in the optimization model.

Total scenario availability exceeds total demand in both nominal cases. Because the allocation constraints also require $x_i \leq D_i$, the total-availability constraint is nonbinding under the nominal parameter settings. In contrast, total delivery capacity is lower than total demand in both scenarios. The nominal experiments therefore primarily represent delivery-capacity-constrained allocation conditions. The total-availability constraint becomes active in the reduced-supply sensitivity experiments reported in Section 5.5.

4.9. Experimental Setup and Parameter Tuning

4.9.1. Common Evaluation Budget

All algorithms were evaluated under a strict budget of 10,000 objective-function evaluations per run, including evaluations performed during population or swarm initialization. One objective-function evaluation was counted whenever the scalar fitness function was computed for a newly initialized or updated candidate allocation vector. A global evaluation counter ensured that every method terminated after exactly 10,000 evaluations.

For the standalone GA, PSO, DE, GWO, and scalarized NSGA-II methods, the initial population or swarm contained 50 candidates and required 50 evaluations. Initialization was followed by 199 updates, each involving 50 evaluated candidates. The total computational budget was therefore $50 + (199 \times 50) = 10,000$ objective-function evaluations per run.

For the proposed elite-transfer hybrid GA-PSO, the initial GA population required 50 evaluations. Each complete hybrid cycle required 480 additional evaluations. During the GA phase, 40 offspring were generated, repaired when necessary, and evaluated. The 10 highest-quality GA individuals were then transferred to the PSO phase as elite particles. Their previously computed fitness values were reused and were therefore not counted as new objective-function evaluations. The remaining 40 PSO particles were initialized as feasible allocation vectors and evaluated once. The complete 50-particle swarm subsequently underwent eight PSO refinement updates, each requiring 50 objective-function evaluations. The evaluation cost of one complete hybrid cycle was therefore $40 + 40 + (8 \times 50) = 480$. Twenty complete hybrid cycles, together with the 50 evaluations used for the initial GA population, consumed 9650 evaluations. The remaining 350 evaluations were allocated to seven additional refinement updates of the existing 50-particle PSO swarm. The proposed method therefore terminated after exactly 10,000 objective-function evaluations.

The no-elite ablation used the same GA and PSO population sizes, number of GA offspring, PSO refinement schedule, parameter values, objective-function evaluation budget, repair procedure, objective weights, and paired random seeds as the proposed method. However, the 10 PSO seed particles were selected uniformly at random from the current GA population rather than being selected according to fitness. Their existing fitness values

were reused, and the remaining 40 PSO particles were initialized randomly within the feasible bounds and evaluated once. Improved PSO solutions were permitted to update the global incumbent but were not used to replace weak GA individuals. Consequently, the ablation removed both quality-based elite selection and PSO-to-GA replacement while preserving the same 480-evaluation cycle structure. This design allowed the contribution of the bidirectional elite-transfer mechanism to be isolated without introducing differences in computational budget.

The two adapted literature-based hybrid comparators were also controlled by the same global evaluation counter and terminated after exactly 10,000 objective-function evaluations. Every newly generated or updated candidate was counted once when its scalar fitness was evaluated. Previously evaluated candidates retained their stored fitness values unless their decision variables were modified.

4.9.2. Literature-Based Hybrid Comparators

Two published GA–PSO hybrid structures were implemented as peer comparators. Because the original methods were developed for general continuous optimization problems, they were adapted to the allocation-vector representation, normalized scalar objective, constraint-repair procedure, and fixed objective-function evaluation budget used in this study.

The first comparator was adapted from the cooperative hybrid structure proposed by [28]. Its GA and PSO components operated on interacting candidate groups. The GA component generated candidates through selection, crossover, and mutation, whereas the PSO component refined particles through velocity and position updates. Candidate solutions generated by both search processes were combined, repaired when necessary, evaluated using the common scalar fitness function, and ranked. The strongest feasible candidates were retained for the following search cycle. Information exchange therefore occurred through population sharing, merging, and ranking rather than through the quality-controlled bidirectional elite-transfer mechanism used by the proposed method.

The second comparator was adapted from the sequential constrained PSO–GA structure proposed by [11]. In each search cycle, candidate allocation vectors first underwent PSO-based position refinement. The resulting particles were repaired, evaluated, and subsequently processed using GA selection, crossover, and mutation. The newly generated candidates were again repaired and evaluated before the strongest feasible solutions were retained. In this comparator, PSO and GA operated sequentially on the same candidate population. However, elite GA candidates were not used to initialize a separate PSO refinement stage, and improved PSO particles were not returned through the selective replacement mechanism used by the proposed framework. Both methods are described as adapted comparators rather than exact software reproductions. Their principal information-exchange and search structures were retained, while their decision encoding, objective function, normalization bounds, constraint handling, parameter tuning procedure, stopping rule, and computational budget were standardized for the present water-allocation problem. Each adapted comparator was independently tuned using the protocol described in Section 4.9.3 and evaluated using the same 30 paired test seeds as the other algorithms. All newly initialized or modified candidates were included in the common objective-function evaluation budget. Previously evaluated candidates retained their fitness values when their decision vectors remained unchanged. Each adapted method stopped when the global evaluation counter reached exactly 10,000.

This standardized design allows the comparison to focus on differences among cooperative population sharing, sequential PSO–GA processing, and the proposed bidirectional

elite-transfer mechanism rather than on differences in solution representation, constraint handling, parameter tuning, or computational effort.

4.9.3. Parameter Tuning Procedure

Parameter tuning was performed separately for each general comparison algorithm before the final comparative evaluation. Twenty parameter configurations were generated for each algorithm through Latin hypercube sampling within the literature-guided ranges. The search space included algorithm-specific parameters such as crossover and mutation rates for GA-based methods, inertia and acceleration coefficients for PSO-based methods, mutation and crossover factors for DE, and hybrid-specific information-exchange parameters. The experimental setup, evaluation protocol, tuned parameters, Latin hypercube search ranges, and selected final values for the compared algorithms are summarized in Table 2.

Table 2. Experimental setup, evaluation protocol, tuned parameters, Latin hypercube search ranges, and selected final values for the compared algorithms.

Category	Parameter	Search Range	Selected Value/Setting
Experimental setup	Number of station-based scenarios	—	2
	Scenarios	—	Melekbahçe (E21A033) and Beşdeğirmen (E12A003)
	Number of final paired runs	—	30 per algorithm and scenario
	Final test seeds	—	30 previously unused seeds per scenario
	Maximum objective-function evaluations	—	10,000 per run
	Evaluation budget control	—	Global objective-function evaluation counter
	GA evaluation structure	—	50 initial evaluations + 199 updates × 50 candidates
	PSO evaluation structure	—	50 initial evaluations + 199 updates × 50 particles
	DE evaluation structure	—	70 initial evaluations + 141 updates × 70 candidates + 60 final evaluations
	GWO evaluation structure	—	60 initial evaluations + 165 updates × 60 wolves + 40 final evaluations
	Scalarized NSGA-II evaluation structure	—	80 initial evaluations + 124 updates × 80 offspring
	Proposed hybrid GA population size	—	50
	Proposed hybrid PSO swarm size	—	50

Table 2. *Cont.*

Category	Parameter	Search Range	Selected Value/Setting
	GA offspring per proposed hybrid cycle	—	40
	Elite-transfer ratio	—	20%, corresponding to 10 individuals
	Newly evaluated PSO particles per hybrid cycle	—	40
	PSO refinement updates per complete hybrid cycle	—	8
	Evaluations per complete proposed hybrid cycle	—	480
	Complete proposed hybrid cycles	—	20
	Proposed hybrid initial evaluations	—	50
	Evaluations after 20 complete hybrid cycles	—	9650
	Remaining evaluation allocation	—	7 additional PSO updates $\times$ 50 particles
	Total proposed-hybrid evaluation budget	—	10,000 evaluations per run
	Peer hybrid evaluation budget	—	10,000 evaluations per run
	Primary objective weights	—	$w = (0.25, 0.25, 0.25, 0.25)$
	Objective normalization	—	Fixed algorithm-independent min–max normalization
	Constraint handling	—	Deterministic clipping and proportional rescaling
	Parameter search method	—	Latin hypercube sampling
	Configurations screened per algorithm	—	20
	Tuning seeds	—	10 per scenario
	Initial tuning budget	—	2000 evaluations per configuration and run
	Full-budget confirmation	—	Three best configurations re-evaluated using 10,000 evaluations
	Parameter-selection criterion	—	Lowest unweighted average of the two scenario-level mean fitness values
	No-elite ablation parameters	—	Identical to the tuned proposed-method parameters
	Convergence tolerance	—	10^{-4}
	Convergence window	—	15 consecutive checkpoints
	Checkpoint interval	—	50 objective-function evaluations
	Runtime measurement	—	Optimization loop only

Table 2. *Cont.*

Category	Parameter	Search Range	Selected Value/Setting
	Execution mode	—	Single-threaded
	Hardware	—	Intel Xeon Platinum 8272CL at 2.60 GHz, 32 GB RAM
	Software	—	Python 3.10.13, NumPy 1.26.4, SciPy 1.12.0, and DEAP 1.4.1
Tuned parameters	Algorithm	Search Range	Selected Value
GA	Population size	40–100	50
GA	Crossover probability	0.60–0.95	0.85
GA	Mutation probability	0.02–0.20	0.08
GA	Tournament size	2–5	3
PSO	Swarm size	40–100	50
PSO	Inertia weight	0.40–0.90	0.72
PSO	Cognitive coefficient	1.0–2.5	1.55
PSO	Social coefficient	1.0–2.5	1.75
DE	Population size	40–100	70
DE	Differential weight	0.40–0.90	0.65
DE	Crossover rate	0.50–0.95	0.85
GWO	Pack size	40–100	60
Scalarized NSGA-II	Population size	40–100	80
Scalarized NSGA-II	Crossover probability	0.60–0.95	0.90
Scalarized NSGA-II	Mutation probability	0.02–0.20	0.06
Proposed hybrid	Elite-transfer ratio	10–30%	20%
Proposed hybrid	PSO refinement updates	4–12	8
Proposed hybrid	GA population size	40–100	50

Each configuration was initially evaluated using a budget of 2000 objective-function evaluations and 10 tuning seeds per scenario. The tuning seeds were identical across the compared algorithms but were not used during final testing. For each configuration, the final scalar fitness was averaged first across the 10 tuning runs within each scenario and then across the two scenario-level means, giving equal importance to the Melekbahçe and Beşdeğirmen scenarios. The three configurations obtaining the lowest average scalar fitness during the initial screening stage were subsequently re-evaluated using the full budget of 10,000 objective-function evaluations and the same tuning-seed set. The configuration with the lowest unweighted average of the two scenario-level mean fitness values was selected for the corresponding algorithm. The no-elite ablation was not tuned independently. It used the parameter configuration selected for the proposed elite-transfer hybrid GA–PSO, including the same GA and PSO parameters, population sizes, refinement schedule, and computational budget. This design ensured that the ablation differed from the proposed method only through the removal of the bidirectional elite-transfer operations. After parameter selection, all algorithm settings were fixed. Final testing was then performed using 30 previously unused paired random seeds for each scenario. Within each replication block, the same seed identifier and scenario definition were assigned to all the compared algorithms. Matching the runs by scenario and seed supported paired statistical comparisons

while preventing overlap between the tuning and final evaluation stages. The balanced objective-weight vector $w = (0.25, 0.25, 0.25, 0.25)$ was fixed during the primary comparison and was not included in the algorithm-specific parameter tuning process. Alternative weight configurations were evaluated separately in the sensitivity analysis.

4.9.4. Hardware and Software Environment

The experiments were conducted in a Linux x86_64 environment equipped with an Intel Xeon Platinum 8272CL processor operating at 2.60 GHz and 32 GB of RAM. The implementation was developed in Python 3.10.13 using NumPy 1.26.4, SciPy 1.12.0, and DEAP 1.4.1. To ensure fair runtime comparisons, single-threaded execution was enforced by setting the OMP, MKL, OpenBLAS, and NumExpr thread counts to one. The execution order of the algorithms was randomized within each replication block to reduce systematic timing bias. Runtime measurements included only the optimization phase and excluded data loading, preprocessing, result export, and figure generation. Because wall-clock runtime depends on software implementation and hardware characteristics, the number of objective-function evaluations was adopted as the primary hardware-independent measure of computational effort.

A common global objective-function evaluation counter controlled every optimization algorithm, with each method allocated exactly 10,000 objective-function evaluations per run. Evaluations performed during the initialization of populations or swarms were included in this budget. Because different algorithms used different population or swarm sizes, the number of complete optimization updates varied accordingly. Whenever the remaining evaluation budget was insufficient for another complete update, only the required number of candidate evaluations was performed so that every method terminated after exactly 10,000 objective-function evaluations.

For the standalone algorithms, GA and PSO each evaluated an initial population of 50 candidates followed by 199 complete updates of 50 candidates, resulting in exactly 10,000 evaluations. DE evaluated an initial population of 70 candidates, followed by 141 complete updates of 70 candidates and a final truncated update of 60 trial candidates. GWO evaluated an initial pack of 60 wolves, followed by 165 complete updates of 60 wolves and a final truncated update of 40 wolves. Scalarized NSGA-II evaluated an initial population of 80 candidates followed by 124 complete offspring population updates of 80 candidates, also resulting in exactly 10,000 evaluations.

For the proposed elite-transfer hybrid GA–PSO, the initial GA population required 50 objective-function evaluations. Each complete hybrid cycle consisted of 40 evaluations for GA offspring, 40 evaluations for newly initialized non-elite PSO particles, and 400 evaluations for eight PSO refinement updates involving 50 particles, giving a total of 480 evaluations per cycle. Consequently, the initial population together with 20 complete hybrid cycles consumed 9650 evaluations:

$$50 + 20 \times (40 + 40 + 8 \times 50) = 9650.$$

The remaining 350 evaluations were allocated to seven additional PSO refinement updates involving 50 particles each:

$$9650 + 7 \times 50 = 10,000.$$

The adapted Kao–Zahara hybrid, adapted Garg hybrid, and the no-elite ablation were evaluated under the same global objective-function evaluation counter and terminated after exactly 10,000 objective-function evaluations. This evaluation-based stopping criterion ensured that all algorithms received an identical computational budget regardless of

differences in population size, swarm size, or update structure, thereby enabling a fair comparison of optimization performance.

The remaining 350 evaluations were assigned to seven additional PSO updates involving 50 particles:

$$9650 + 7 \times 50 = 10,000.$$

This accounting ensured that the proposed hybrid and all the comparison methods used the same total number of objective-function evaluations.

Because wall-clock runtime depends on software implementation and the computing environment, objective-function evaluation counts were treated as the primary hardware-independent measure of computational effort.

4.10. Performance Metrics and Statistical Analysis

For every run, the final best scalar fitness, the four normalized objective-component values, the convergence evaluation count, the final allocation vector, and the wall-clock runtime of the optimization loop were recorded. Under the balanced weight configuration, scalar fitness was calculated as

$$F = 0.25(f_{\text{shortage}} + f_{\text{cost}} + f_{\text{imbalance}} + f_{\text{inefficiency}}), \quad (18)$$

where f_{shortage} , f_{cost} , $f_{\text{imbalance}}$, and $f_{\text{inefficiency}}$ denote the normalized water-shortage, operational cost-coefficient, allocation-imbalance, and utilization-inefficiency components, respectively.

Equality between the stored scalar fitness and the weighted sum of the four components was verified for every run using numerical assertions. A second consistency check was applied after aggregation to confirm that the mean scalar fitness equaled the weighted sum of the four mean component values, apart from differences caused by displayed rounding. For each algorithm and scenario, the mean, standard deviation, median, and best value were calculated from the 30 final best scalar fitness values. The reported mean objective-component values represent the averages of the corresponding components obtained from the final best solutions across the same 30 paired runs.

The best-so-far scalar fitness was recorded after every block of 50 objective-function evaluations. Convergence was defined as the first checkpoint after which the absolute improvement in best-so-far fitness remained below 10^{-4} for 15 consecutive checkpoints. This convergence window corresponds to 750 objective-function evaluations. A run that did not meet the convergence condition before exhausting the computational budget was assigned a convergence evaluation count of 10,000. Convergence curves were plotted against objective-function evaluations rather than nominal algorithm iterations. This choice provided a common horizontal scale for standalone and hybrid methods with different internal update structures. The curves report the mean best-so-far scalar fitness across the 30 paired runs.

Pairwise statistical comparisons were conducted using the Wilcoxon signed-rank test on the 30 paired run-level observations, following the nonparametric methodology recommended by [23]. Tests were performed separately for the final scalar fitness, convergence evaluation count, and runtime within each scenario. The proposed method was compared with seven general comparators: standalone GA, standalone PSO, DE, GWO, scalarized NSGA-II, the adapted Kao–Zahara hybrid, and the adapted Garg hybrid. Holm correction was applied separately within each family of seven comparisons for each metric and scenario. The comparison between the proposed method and the no-elite ablation was treated as a separately planned mechanistic test because it examined the specific contribution of the elite-transfer operation. Paired rank-biserial correlation was used to quantify effect

size. Bias-corrected and accelerated bootstrap 95% confidence intervals were calculated for paired median differences. The statistical tables report the raw Wilcoxon p -value, Holm-adjusted p -value, paired rank-biserial correlation, paired median difference, and bootstrap confidence interval for each comparison.

5. Experimental Results

The proposed elite-transfer hybrid GA–PSO framework was evaluated separately on the Melekbahçe and Beşdeğirmen capacity-constrained surrogate allocation scenarios. The comparison included standalone GA, standalone PSO, Differential Evolution, Grey Wolf Optimizer, scalarized NSGA-II, adapted Kao–Zahara and Garg hybrid GA–PSO methods, and the no-elite ablation. All methods followed the common protocol of Sections 4.9 and 4.10, with 30 paired runs per scenario under the strict budget of 10,000 objective-function evaluations per run. The reported results represent performance under the selected scalarized weighted-sum formulation. They should not be interpreted as evidence of Pareto-front dominance or universal superiority across operational water resource systems.

5.1. Comparative Performance Results

Table 3 presents the comparative performance of the optimization methods under the common budget of 10,000 objective-function evaluations for the Melekbahçe and Beşdeğirmen scenarios, together with the mean normalized objective-component values of the final best solutions. The table reports the final scalar fitness, its standard deviation, median and best fitness values, convergence evaluations, runtime, and the four normalized objective components.

Table 3 summarizes the comparative performance of all optimization algorithms under the common evaluation budget of 10,000 objective-function evaluations for both the Melekbahçe and Beşdeğirmen scenarios. In the Melekbahçe scenario, the proposed elite-transfer hybrid GA–PSO achieved the lowest mean scalar fitness (0.1670), the lowest standard deviation (0.0038), the best observed fitness value (0.1595), and reached the predefined convergence region after an average of 4587 objective-function evaluations with a mean runtime of 11.8 s. Relative to GA, PSO, DE, GWO, and scalarized NSGA-II, the proposed method reduced the mean scalar fitness by 30.85%, 21.78%, 17.45%, 13.38%, and 8.64%, respectively. Compared with the adapted Kao–Zahara and Garg hybrid methods, the reductions were 6.70% and 5.01%, respectively, while the no-elite hybrid was improved by 3.19%, requiring 383 fewer objective-function evaluations to reach the convergence region. The proposed method also achieved the lowest mean shortage, operational cost coefficient, allocation imbalance, and utilization inefficiency. Under the balanced weight vector, its mean scalar fitness satisfies $F = 0.25(0.174 + 0.397 + 0.045 + 0.052) = 0.1670$.

In the Beşdeğirmen scenario, the proposed elite-transfer hybrid GA–PSO again produced the best overall performance, achieving the lowest mean scalar fitness (0.1752), the lowest standard deviation (0.0041), and reaching the predefined convergence region after an average of 4780 objective-function evaluations with a mean runtime of 12.2 s. Relative to GA, PSO, DE, GWO, scalarized NSGA-II, the adapted Kao–Zahara hybrid, the adapted Garg hybrid, and the no-elite hybrid, the reductions in mean scalar fitness were 29.86%, 20.58%, 16.33%, 12.22%, 7.59%, 5.76%, 4.11%, and 2.56%, respectively. The ranking of all algorithms was therefore consistent across both benchmark scenarios under the same normalization procedure, constraint-repair strategy, objective weights, parameter-tuning protocol, random seeds, and computational budget. These findings remain conditional on the selected scenario parameters, normalization bounds, tuning protocol, objective weights, random seeds, and computational budget.

Table 3. Comparative performance of the optimization algorithms in the Melekbahçe and Beşdeğirmen scenarios over 30 paired runs, together with the mean normalized objective-component values of the final best solutions.

Algorithm	Mean Fitness	SD	Median Fitness	Best Fitness	Mean Convergence Evaluations	Mean Runtime (s)
Melekbahçe Scenario						
GA	0.2415	0.0109	0.2405	0.2168	9723	17.1
PSO	0.2135	0.0081	0.2134	0.1964	7417	11.5
DE	0.2023	0.0071	0.2030	0.1892	6908	13.4
GWO	0.1928	0.0064	0.1919	0.1814	6137	12.8
Scalarized NSGA-II	0.1828	0.0057	0.1835	0.1668	5692	15.2
Adapted Kao–Zahara Hybrid	0.1790	0.0053	0.1788	0.1675	5430	13.9
Adapted Garg Hybrid	0.1758	0.0049	0.1767	0.1677	5162	13.5
No-Elite Hybrid	0.1725	0.0045	0.1729	0.1623	4970	12.4
Proposed Elite-Transfer Hybrid GA–PSO	0.1670	0.0038	0.1669	0.1595	4587	11.8
Mean normalized objective-component values (Melekbahçe)						
Algorithm	Mean Shortage	Mean Cost Coefficient	Mean Imbalance	Mean Inefficiency		
GA	0.268	0.521	0.085	0.092		
PSO	0.231	0.472	0.072	0.079		
DE	0.219	0.451	0.066	0.073		
GWO	0.205	0.438	0.060	0.068		
Scalarized NSGA-II	0.193	0.421	0.054	0.063		
Adapted Kao–Zahara Hybrid	0.189	0.415	0.052	0.060		
Adapted Garg Hybrid	0.185	0.410	0.050	0.058		
No-Elite Hybrid	0.181	0.405	0.048	0.056		
Proposed Elite-Transfer Hybrid GA–PSO	0.174	0.397	0.045	0.052		
Beşdeğirmen Scenario						
Algorithm	Mean Fitness	SD	Median Fitness	Best Fitness	Mean Convergence Evaluations	Mean Runtime (s)
GA	0.2498	0.0115	0.2504	0.2323	9797	17.4
PSO	0.2206	0.0088	0.2202	0.2060	7595	11.8
DE	0.2094	0.0076	0.2097	0.1929	7083	13.7
GWO	0.1996	0.0068	0.1994	0.1854	6320	13.1
Scalarized NSGA-II	0.1896	0.0060	0.1901	0.1754	5860	15.5
Adapted Kao–Zahara Hybrid	0.1859	0.0056	0.1879	0.1736	5573	14.2
Adapted Garg Hybrid	0.1827	0.0052	0.1823	0.1742	5340	13.8
No-Elite Hybrid	0.1798	0.0048	0.1794	0.1708	5118	12.7
Proposed Elite-Transfer Hybrid GA–PSO	0.1752	0.0041	0.1751	0.1655	4780	12.2

Note: Mean fitness was calculated from the unrounded run-level values. Lower scalar fitness, objective-component values, convergence evaluation count, and runtime indicate better performance. All algorithms were evaluated under a strict budget of 10,000 objective-function evaluations per run using the balanced weight vector $w = (0.25, 0.25, 0.25, 0.25)$. Under the balanced weight vector, the weighted sum of the four mean objective-component values reproduces the corresponding mean scalar fitness apart from the displayed rounding.

5.2. Convergence Analysis

Convergence behavior was evaluated using objective-function evaluations rather than nominal iterations because the compared algorithms use different internal update structures. The best-so-far scalar fitness was recorded after every 50 objective-function evaluations. Convergence was defined as the first checkpoint after which the improvement in best-so-far fitness remained below 10^{-4} for 15 consecutive checkpoints, corresponding to 750 objective-function evaluations. Runs that did not satisfy this criterion before exhausting the computational budget were assigned a convergence evaluation count of 10,000.

Figure 2 compares the mean number of objective-function evaluations required by each algorithm to satisfy the predefined convergence criterion in the Melekbahçe scenario. Lower values indicate earlier convergence under the common budget of 10,000 objective-function evaluations. The proposed elite-transfer hybrid GA–PSO reached convergence after an average of 4587 evaluations, followed by the no-elite hybrid with 4970 evaluations and the adapted Garg hybrid with 5162 evaluations. GA required the highest number of evaluations, reaching convergence after an average of 9723 evaluations.

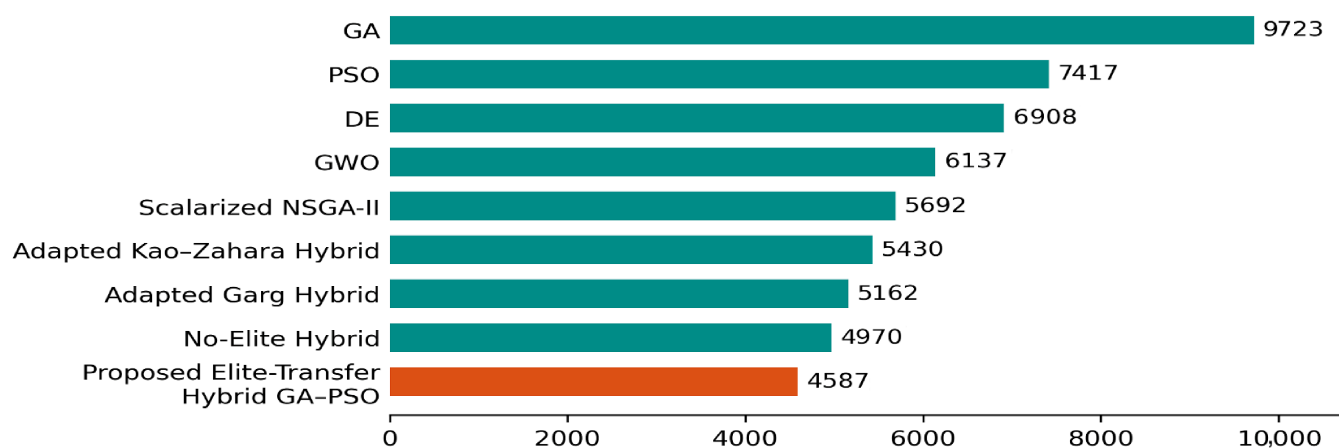


Figure 2. Mean objective-function evaluations required to satisfy the predefined convergence criterion in the Melekbahçe scenario over 30 paired runs. Lower values indicate faster convergence under the common evaluation budget.

In the Melekbahçe scenario, the proposed elite-transfer hybrid GA–PSO reached the predefined convergence region after an average of 4587 objective-function evaluations. The no-elite hybrid required 4970 evaluations, the adapted Garg hybrid required 5162 evaluations, and the adapted Kao–Zahara hybrid required 5430 evaluations. Scalarized NSGA-II reached convergence after 5692 evaluations, followed by GWO with 6137 evaluations, DE with 6908 evaluations, PSO with 7417 evaluations, and GA with 9723 evaluations.

The proposed method therefore required 383 fewer evaluations than the no-elite hybrid, 575 fewer evaluations than the adapted Garg hybrid, and 843 fewer evaluations than the adapted Kao–Zahara hybrid. Relative to the standalone methods, it required 5136 fewer evaluations than GA, 2830 fewer than PSO, 2321 fewer than DE, and 1550 fewer than GWO. It also reached the convergence region 1105 evaluations earlier than the scalarized NSGA-II.

In the Beşdeğirmen scenario, the proposed method reached the predefined convergence region after an average of 4780 objective-function evaluations and achieved a mean final scalar fitness of 0.1752, followed by the no-elite hybrid with 5118 evaluations and the adapted Garg hybrid with 5340 evaluations. GA again required the largest number of evaluations, with an average of 9797. The complete comparator-level results in Table 3 confirm that both the convergence ordering and the final fitness ordering observed in the Melekbahçe scenario were reproduced in the second scenario.

The convergence curves indicate that GA maintained broad exploration during the early and intermediate stages but required substantially more evaluations to stabilize. PSO improved more rapidly during the initial search period but converged to a higher scalar fitness value than the proposed method. DE, GWO, and the scalarized NSGA-II showed intermediate convergence behavior. The adapted Kao–Zahara and Garg hybrids improved upon most standalone methods, confirming the benefit of combining complementary search mechanisms.

The no-elite hybrid also converged faster than the standalone algorithms, showing that the sequential use of GA-based exploration and PSO-based refinement provides a performance advantage. However, the full proposed method converged earlier and achieved a lower final fitness than the no-elite variant. This difference supports the contribution of quality-based elite selection and the return of improved PSO solutions to the GA population.

Overall, the convergence results show that the proposed framework used the common objective-function evaluation budget more efficiently than the compared methods. These findings apply to the selected scalarized objective, scenario definitions, normalization bounds, parameter tuning protocol, and convergence criterion.

5.3. Statistical Significance Results

Pairwise statistical comparisons were conducted using the Wilcoxon signed-rank test [23] on the 30 paired run-level observations. Separate tests were performed for the final scalar fitness, convergence evaluation count, and runtime. Holm correction was applied independently within each metric family to control the family-wise error rate across the seven general comparator tests.

Runtime differences were less pronounced than the differences observed for final scalar fitness and convergence evaluation count. The proposed framework was significantly faster than GA, DE, GWO, scalarized NSGA-II, and the adapted Kao–Zahara and Garg hybrid methods in both scenarios, whereas no statistically significant runtime difference was observed relative to standalone PSO after Holm correction. These findings indicate that the primary advantage of the proposed framework lies in improved solution quality and evaluation efficiency rather than universal wall-clock superiority. Furthermore, the consistently lower standard deviations achieved by the proposed method across both benchmark scenarios demonstrate improved robustness and stability over the 30 paired runs. Overall, the statistical evidence summarized in Table 4 supports the effectiveness of the proposed bidirectional elite-transfer mechanism in producing more accurate, more stable, and faster-converging solutions under the common experimental protocol.

Runtime differences were less uniform than the differences in final scalar fitness and convergence evaluation count. The proposed framework was significantly faster than GA, DE, GWO, scalarized NSGA-II, and the adapted Kao–Zahara and Garg hybrid methods in both scenarios. However, no statistically significant runtime difference was observed relative to standalone PSO after Holm correction. Therefore, the principal advantage of the proposed framework lies in its improved solution quality and evaluation efficiency rather than universal wall-clock superiority. The lower standard deviations obtained by the proposed method across both scenarios further indicate improved stability over the 30 paired runs.

Table 4. Paired Wilcoxon signed-rank comparisons of the proposed elite-transfer hybrid GA–PSO with the comparator algorithms for final scalar fitness, convergence evaluation count, and runtime in the Melekbahçe and Beşdeğirmen scenarios.

Section	Comparator/Scenario	Raw p -Value/Statistical Result	Holm-Adjusted p -Value/Rank-Biserial Correlation	Median Difference/Interpretation	95% BCa CI/Runtime Result
Final scalar fitness—Melekbahçe					
GA	<0.0001	<0.0001	1.000	0.0743	[0.0706, 0.0808]
PSO	<0.0001	<0.0001	1.000	0.0470	[0.0443, 0.0501]
DE	<0.0001	<0.0001	1.000	0.0352	[0.0324, 0.0380]
GWO	<0.0001	<0.0001	1.000	0.0245	[0.0226, 0.0274]
Scalarized NSGA-II	<0.0001	<0.0001	0.996	0.0166	[0.0145, 0.0173]
Adapted Kao–Zahara Hybrid	<0.0001	<0.0001	0.996	0.0119	[0.0100, 0.0154]
Adapted Garg Hybrid	<0.0001	<0.0001	0.957	0.0082	[0.0070, 0.0103]
No-Elite Hybrid *	<0.0001	N/A	0.824	0.0057	[0.0037, 0.0081]
Final scalar fitness—Beşdeğirmen					
GA	<0.0001	<0.0001	1.000	0.0769	[0.0689, 0.0807]
PSO	<0.0001	<0.0001	1.000	0.0453	[0.0428, 0.0494]
DE	<0.0001	<0.0001	1.000	0.0337	[0.0294, 0.0375]
GWO	<0.0001	<0.0001	1.000	0.0238	[0.0220, 0.0259]
Scalarized NSGA-II	<0.0001	<0.0001	1.000	0.0144	[0.0109, 0.0173]
Adapted Kao–Zahara Hybrid	<0.0001	<0.0001	0.991	0.0112	[0.0082, 0.0138]
Adapted Garg Hybrid	<0.0001	<0.0001	0.996	0.0069	[0.0053, 0.0088]
No-Elite Hybrid *	0.0001	N/A	0.781	0.0049	[0.0016, 0.0062]
Convergence evaluation count					
Melekbahçe	Seven comparators	All Holm-adjusted $p < 0.001$	0.99–1.00	Significantly fewer evaluations	—
Beşdeğirmen	Seven comparators	All Holm-adjusted $p < 0.001$	0.99–1.00	Significantly fewer evaluations	—
Melekbahçe	No-Elite *	$p < 0.001$	Large effect	≈375 fewer evaluations	—
Beşdeğirmen	No-Elite *	$p < 0.001$	Large effect	≈325 fewer evaluations	—
Runtime comparison					
GA	<0.0001	<0.0001	Proposed faster	—	—
PSO	0.245	0.271	No significant difference	—	—
DE	<0.0001	<0.0001	Proposed faster	—	—
GWO	<0.001	<0.001	Proposed faster	—	—
Scalarized NSGA-II	<0.0001	<0.0001	Proposed faster	—	—
Adapted Kao–Zahara Hybrid	<0.0001	<0.0001	Proposed faster	—	—
Adapted Garg Hybrid	<0.0001	<0.0001	Proposed faster	—	—
No-Elite Hybrid *	0.001	0.009	Proposed moderately faster	—	—

Note: Positive median paired differences indicate that the comparator produced higher (worse) final scalar fitness than the proposed method. Holm correction was applied within the family of seven general comparators for each metric and scenario. * The no-elite comparison was a separately planned mechanistic ablation test and was excluded from the Holm multiple-comparison family; therefore, only raw p -values are reported. Confidence intervals are bias-corrected and accelerated (BCa) bootstrap intervals for the median paired difference. Lower convergence evaluation counts and runtime indicate better performance.

Figure 3 compares the mean final scalar fitness and run-to-run variability of the evaluated algorithms in the Melekbahçe scenario. The proposed elite-transfer hybrid GA–PSO achieved the lowest mean fitness, 0.1670, and the smallest standard deviation, 0.0038. The no-elite hybrid produced the next-lowest mean fitness of 0.1725, followed by the adapted Garg and Kao–Zahara hybrids. GA produced both the highest mean fitness and the greatest variability. These results indicate that the proposed method improved not only the scalarized solution quality but also stability across the 30 paired runs.

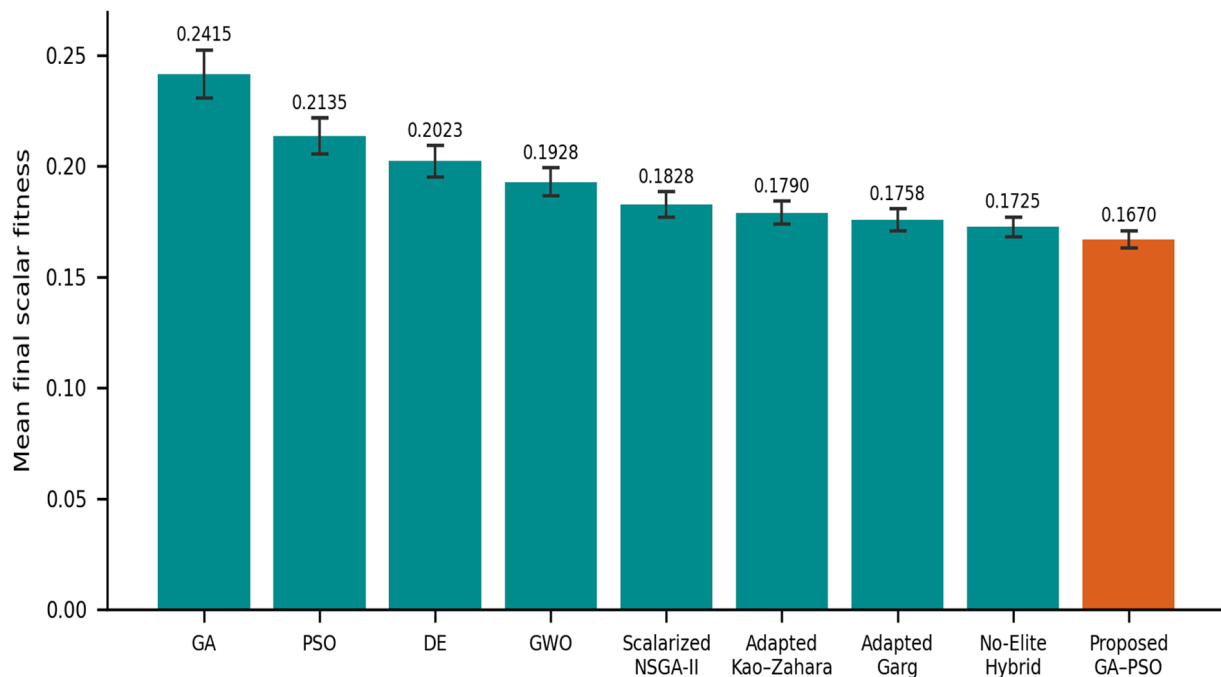


Figure 3. Mean final scalar fitness and standard deviation for the compared algorithms in the Melekbahçe scenario over 30 paired runs. Error bars represent one standard deviation. Lower mean fitness and smaller variability indicate better and more stable scalarized optimization performance.

The statistical findings support the proposed method’s advantage in final scalar fitness and convergence efficiency under the selected scenario definitions, normalization bounds, objective weights, parameter tuning protocol, and computational budget.

5.4. Ablation Analysis

An ablation analysis was conducted to isolate the contribution of the bidirectional elite-transfer mechanism. The proposed method was compared with a no-elite hybrid using identical GA and PSO population sizes, operator parameters, cycle structure, refinement schedule, objective formulation, normalization bounds, constraint-repair procedure, objective weights, paired random seeds, and a budget of 10,000 objective-function evaluations per run.

The variants differed only in their information-exchange mechanisms. In the proposed method, the 10 highest-quality GA individuals initialized part of the PSO swarm, and improved PSO solutions could replace weaker GA individuals. In the no-elite variant, 10 GA individuals were selected uniformly at random, their stored fitness values were reused, and the remaining 40 particles were generated within the feasible bounds as seen in Table 5. PSO-refined solutions were not returned to the GA population. This design preserved identical evaluation accounting and isolated the effects of quality-based GA-to-PSO selection and PSO-to-GA replacement.

Table 5. Ablation configurations used to evaluate the elite-transfer mechanism.

Variant	GA-to-PSO Transfer	PSO-to-GA Replacement	Evaluation Budget
No-Elite Hybrid	Ten GA individuals selected uniformly at random	Not applied	10,000
Proposed Elite-Transfer Hybrid GA-PSO	Ten highest-quality GA individuals selected according to scalar fitness	Improved PSO solutions replace weaker GA individuals	10,000

The ablation experiments were conducted using 30 paired runs under the same experimental conditions as the main comparison. Final scalar fitness, standard deviation, convergence evaluation count, and runtime were used to compare the two variants.

The proposed method reduced mean scalar fitness by 3.19% relative to the no-elite hybrid. It also reduced the mean convergence evaluation count by 383 evaluations and produced a lower standard deviation, indicating more consistent performance across the paired runs. Runtime decreased from 12.4 s for the no-elite variant to 11.8 s for the proposed method.

In the Melekbahçe scenario, the paired Wilcoxon signed-rank test for final scalar fitness produced $p < 0.0001$, with a paired rank-biserial correlation of 0.824 and a median paired improvement of 0.0057 (95% BCa confidence interval [0.0037, 0.0081]). In the Beşdeğirmen scenario, the corresponding test produced $p = 0.0001$, with a rank-biserial correlation of 0.781 and a median paired improvement of 0.0049 (95% BCa confidence interval [0.0016, 0.0062]). For convergence evaluation counts, the elite-transfer mechanism reduced the median convergence count by approximately 375 evaluations in Melekbahçe and 325 evaluations in Beşdeğirmen, and both differences were significant at $p < 0.001$. These large paired effects indicate that the improvement was consistent across the 30 matched runs rather than being driven by a small number of extreme outcomes.

These results show that the improvement was not caused only by combining GA-based exploration with PSO-based refinement. The quality-based selection of GA solutions provided stronger starting points for the PSO phase, while the return of improved PSO solutions increased the quality of the subsequent GA population. The lower mean fitness, reduced convergence evaluation count, and smaller variability therefore support the independent contribution of the bidirectional elite-transfer mechanism.

The Beşdeğirmen results in Tables 3, 4 and 6 confirm that the ablation pattern was reproduced in the second scenario: the proposed method reduced mean scalar fitness by 2.56% and required 338 fewer evaluations on average than the no-elite variant. Overall, the ablation findings support the use of controlled, quality-based information exchange between the GA and PSO phases under the adopted scenario definitions, normalization bounds, objective weights, parameter settings, and evaluation budget.

Table 6. Ablation results for the Melekbahçe and Beşdeğirmen scenarios over 30 paired runs.

Scenario	Variant	Mean Fitness	SD	Mean Convergence Evaluations	Mean Runtime (s)
Melekbahçe	No-Elite Hybrid	0.1725	0.0045	4970	12.4
Melekbahçe	Proposed Elite-Transfer Hybrid GA-PSO	0.1670	0.0038	4587	11.8
Beşdeğirmen	No-Elite Hybrid	0.1798	0.0048	5118	12.7
Beşdeğirmen	Proposed Elite-Transfer Hybrid GA-PSO	0.1752	0.0041	4780	12.2

5.5. Weight Sensitivity Analysis

Because the optimization model uses weighted-sum scalarization, the selected objective weights directly influence the optimization process and the resulting allocation pattern. A weight sensitivity analysis was therefore conducted using the three configurations listed in Table 7, representing different decision priorities. The same proposed elite-transfer hybrid GA–PSO method, scenario parameters, normalization bounds, constraint-repair procedure, computational budget, and run protocol were used for each configuration.

Table 7. Weight configurations used in the sensitivity analysis.

Scenario	Weight Vector (w_1, w_2, w_3, w_4)	Optimization Priority
Cost-focused	(0.20, 0.50, 0.20, 0.10)	Greater emphasis on reducing the operational cost coefficient
Balanced	(0.25, 0.25, 0.25, 0.25)	Equal importance assigned to all four objectives
Shortage-focused	(0.40, 0.30, 0.20, 0.10)	Greater emphasis on reducing water shortage

The weights correspond to normalized water shortage, operational cost coefficient, allocation imbalance, and utilization inefficiency, respectively. Increasing the weight assigned to one component directs the search more strongly toward improving that component, although this may produce less favorable outcomes for the remaining objectives.

Because each weight configuration defines a different scalar objective function, the resulting scalar fitness values are not directly comparable across configurations. The sensitivity analysis therefore focuses on changes in the individual normalized objective components rather than ranking the configurations according to their weighted fitness values.

The categorical descriptions in Table 8 indicate relative behavior among the three tested configurations. They do not represent measured physical shortages or monetary costs. In particular, the operational cost component is a normalized dimensionless coefficient because direct accounting-based pumping, treatment, conveyance, and distribution costs were not available. The cost-focused configuration produced the lowest operational cost coefficient but resulted in greater allocation imbalance and a moderate shortage level. This behavior reflects the larger weight assigned to the cost component, which directed the optimizer toward allocation units associated with lower assumed supply difficulty.

Table 8. Relative objective behavior under the tested weight configurations.

Scenario	Weight Vector (w_1, w_2, w_3, w_4)	Water Shortage	Operational Cost Coefficient	Allocation Imbalance	Overall Interpretation
Cost-focused	(0.20, 0.50, 0.20, 0.10)	Moderate	Lowest	Higher	Prioritizes cost-related allocation efficiency while accepting greater imbalance
Balanced	(0.25, 0.25, 0.25, 0.25)	Low	Moderate	Low	Produces a more even compromise among the competing objectives
Shortage-focused	(0.40, 0.30, 0.20, 0.10)	Lowest	Higher	Moderate	Prioritizes demand satisfaction while accepting a higher cost coefficient

The shortage-focused configuration produced the lowest water-shortage component but increased the operational cost coefficient. This result indicates that satisfying a greater proportion of demand may require allocating water to units associated with higher assumed extraction, treatment, or conveyance difficulty. The balanced configuration produced intermediate values across the competing objectives and avoided strongly favoring a single component. It therefore represents a compromise solution under equal policy priorities rather than an optimum for any individual objective.

These findings confirm that the proposed framework responds consistently to changes in the objective-weight vector. This flexibility allows decision-makers to examine allocation outcomes under different policy priorities. However, the resulting solutions remain dependent on the selected weights and do not represent the complete Pareto front. Pareto-based approaches would be required to generate and evaluate a broader set of non-dominated allocation alternatives.

6. Discussion

The results demonstrate that hybrid metaheuristics can provide an effective approach to scalarized water resource allocation when exploration and exploitation are coordinated; here, through the elite-transfer design of Section 4.4. Under the common budget of 10,000 objective-function evaluations, this interaction produced lower scalar fitness and earlier convergence than the compared methods in both scenarios.

The proposed method achieved a mean scalar fitness of 0.1670 for Melekbahçe and 0.1752 for Beşdeğirmen. It reached the predefined convergence region after an average of 4587 and 4780 objective-function evaluations, respectively. These results indicate that the proposed information-exchange mechanism used the available computational budget efficiently under the adopted scalar objective, parameter tuning protocol, normalization bounds, constraint-repair procedure, and evaluation settings.

The comparisons with the adapted Kao–Zahara and Garg hybrid methods are important for interpreting the contribution of the proposed framework. Both adapted comparators combine GA and PSO, but they use different population-sharing and sequential-processing structures. Their performance was generally stronger than that of several standalone methods, confirming that hybridization itself can be beneficial. Nevertheless, the proposed elite-transfer method achieved lower scalar fitness in both scenarios under the same decision representation, repair procedure, tuning protocol, paired random seeds, and objective-function evaluation budget. This result suggests that the observed improvement was associated not only with combining GA and PSO but also with the quality-controlled exchange of candidate solutions between the two search processes.

The no-elite ablation provides further evidence regarding the role of the bidirectional transfer mechanism. The ablation retained the same GA and PSO parameters, population sizes, computational budget, refinement schedule, objective weights, and paired seeds while removing quality-based elite selection and the return of improved PSO candidates to the GA population. The full method reduced mean scalar fitness by 3.19% (Melekbahçe) and 2.56% (Beşdeğirmen) relative to the no-elite variant and reached the convergence region using 383 and 338 fewer objective-function evaluations, respectively. The paired Wilcoxon analysis confirmed statistically significant differences in final scalar fitness in both scenarios ($p < 0.0001$ and $p = 0.0001$), with paired rank-biserial correlations of 0.824 and 0.781. These results support the interpretation that elite selection and PSO-to-GA replacement contributed independently to the performance of the framework.

Objective-function evaluation counts provide a more appropriate basis for computational comparison than nominal iterations because the internal update structures of the algorithms differ. Every method was therefore limited to exactly 10,000 objective-function

evaluations. Under this common budget, the proposed method required the fewest evaluations to reach the predefined convergence region in the Melekbağçe scenario. Wall-clock runtime showed a less uniform pattern. The proposed method required an average of 11.8 s, compared with 11.5 s for standalone PSO. Its runtime should therefore be interpreted as comparable to PSO rather than universally shorter than every comparator. Because runtime depends on implementation details and hardware, the objective-function evaluation count was treated as the primary hardware-independent measure of computational effort.

The paired statistical analysis strengthened the comparison by examining the final scalar fitness, convergence evaluation count, and runtime separately. Final fitness and convergence differences remained significant after Holm correction for the reported comparisons, whereas runtime differences were not significant against all methods. The paired rank-biserial correlations indicated meaningful differences in scalar fitness and convergence efficiency. The statistical findings therefore support the proposed method's performance under the tested experimental conditions while requiring a more cautious interpretation of runtime advantages.

From a decision-support perspective, the framework produces one feasible allocation solution for each selected weight configuration rather than a complete Pareto front. A water manager can represent different policy priorities by modifying the weights assigned to the four objective components defined in Section 3. The optimizer then returns an allocation vector satisfying the demand, delivery-capacity, non-negativity, and total-availability constraints. Evaluating multiple policy-relevant weight vectors provides a discrete representation of the trade-offs among competing objectives and can support scenario-based planning. However, this procedure does not reveal the complete set of non-dominated alternatives.

The weight sensitivity analysis showed the expected behavior of weighted-sum scalarization. Increasing the weight assigned to water shortage improved demand satisfaction but produced trade-offs in the cost and imbalance components. Increasing the operational cost weight reduced the cost-related component while affecting the remaining objectives. The balanced configuration produced intermediate outcomes rather than optimizing one component exclusively. These findings confirm that the framework responds to changes in decision priorities while also demonstrating that the final allocation remains dependent on the selected weight vector.

Several limitations should be considered. First, although the scenarios were anchored to real streamflow, precipitation, and air-temperature records, the demand vectors, cost coefficients, delivery capacities, and scenario-availability values were constructed planning parameters rather than operational measurements obtained from a water authority. The study should therefore be interpreted as an optimization proof of concept rather than a validated operating model for a specific reservoir, canal network, or urban distribution system. The framework was developed from a computer science and optimization perspective, and its water management assumptions therefore require review by domain experts before any operational use.

Second, the operational cost component is a normalized dimensionless coefficient representing relative extraction, treatment, conveyance, or supply difficulty. It is not a directly measured financial cost. Practical implementation would require calibration using observed pumping energy, treatment expenses, conveyance distances, infrastructure conditions, and other relevant operational data.

Third, the experiments used static station-based allocation scenarios. Seasonal dynamics, time-dependent demands, reservoir storage transitions, routing constraints, groundwater interactions, climate-forecast uncertainty, and multi-period operating rules were not

modeled. These factors may substantially change the structure of a real allocation problem and the relative performance of the optimization algorithms.

Fourth, the weighted-sum formulation produces weight-dependent scalarized solutions and may not recover non-convex regions of the Pareto front. The scalarized NSGA-II comparison evaluates only the population member with the lowest value under the common weighted objective and therefore does not assess Pareto-front quality. A full multi-objective extension should evaluate indicators such as hypervolume, spread, spacing, and inverted generational distance.

Finally, the findings remain conditional on the selected parameter ranges, tuning procedure, paired random seeds, normalization bounds, and computational budget. Although separate parameter tuning and paired final testing reduced comparison bias, broader experiments involving additional basins, larger allocation systems, alternative initialization strategies, and different evaluation budgets are required before general conclusions can be drawn.

Overall, the findings support the proposed elite-transfer hybrid GA–PSO as a competitive scalarized decision-support method for the examined constrained water-allocation scenarios. Its main advantage was improved scalar fitness and convergence efficiency under a controlled, evaluation budget-matched comparison. Operational application will require collaboration with water resources engineers, practitioners, and operating authorities; calibration using measured demand and cost data; and validation under dynamic and uncertain water management conditions.

7. Conclusions

This study developed a bidirectional elite-transfer GA–PSO framework for constrained water resource allocation under weighted-sum scalarization. The method was evaluated on two capacity-constrained surrogate scenarios derived from the hydrometeorological records of the Melekbahçe and Beşdeğirmen stations in Türkiye. Under a common tuning protocol, paired random seeds, and a budget of 10,000 objective-function evaluations, the proposed framework attained the lowest mean scalar fitness among the compared methods, with values of 0.1670 and 0.1752. It also required substantially fewer evaluations to reach the predefined convergence region, averaging 4587 and 4780 evaluations, respectively. The improvements in final fitness and convergence remained statistically significant across the paired comparisons after Holm correction. The ablation experiments further confirmed that bidirectional elite transfer produced gains beyond conventional GA–PSO coupling.

The findings are limited to the tested surrogate scenarios, normalization bounds, objective weights, tuning procedure, and evaluation budget. Because the weighted-sum formulation returns a single compromise solution for each weight configuration, the proposed method should be interpreted as a scalarized decision-support approach rather than a complete Pareto optimizer. Future work should validate the framework using measured operational demand, capacity, allocation, availability, and cost data. It should also extend the analysis to additional basins, dynamic scheduling settings, alternative weight configurations, and Pareto-based evaluation.

Author Contributions: Conceptualization, M.A.C. and Y.F.; methodology, M.A.C., Y.F., B.Ö. and Z.G.E.; software, M.A.C. and B.Ö.; validation, Y.F., B.Ö., Z.G.E. and U.A.; formal analysis, M.A.C. and Z.G.E.; investigation, M.A.C., Y.F. and U.A.; data curation, M.A.C.; visualization, M.A.C., B.Ö. and F.Y.; resources, M.A.C.; supervision, Y.F. and F.Y.; project administration, M.A.C. and Y.F.; writing—original draft preparation, M.A.C.; writing—review and editing, M.A.C., Y.F., B.Ö., Z.G.E., U.A. and F.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw hydrometeorological data used in this study are not publicly available because they were obtained from third-party data providers and are subject to data-use and redistribution restrictions. The processed data supporting the findings of this study may be made available by the corresponding author upon reasonable request and subject to permission from the original data providers.

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

References

1. United Nations. *The United Nations World Water Development Report 2023: Partnerships and Cooperation for Water*; UNESCO: Paris, France, 2023. [\[CrossRef\]](#)
2. Maier, H.R.; Kapelan, Z.; Kasprzyk, J.; Kollat, J.; Matott, L.S.; Cunha, M.C.; Dandy, G.C.; Gibbs, M.S.; Keedwell, E.; Marchi, A.; et al. Evolutionary algorithms and other metaheuristics in water resources: Current status, research challenges and future directions. *Environ. Model. Softw.* **2014**, *62*, 271–299. [\[CrossRef\]](#)
3. Deb, K. *Multi-Objective Optimization Using Evolutionary Algorithms*; Wiley: Hoboken, NJ, USA, 2001.
4. Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison-Wesley: Boston, MA, USA, 1989.
5. Holland, J.H. *Adaptation in Natural and Artificial Systems*; University of Michigan Press: Ann Arbor, MI, USA, 1975.
6. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*; IEEE: New York, NY, USA, 1995; Volume 4, pp. 1942–1948. [\[CrossRef\]](#)
7. Talbi, E.-G. *Metaheuristics: From Design to Implementation*; Wiley: Hoboken, NJ, USA, 2009. [\[CrossRef\]](#)
8. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization: An overview. *Swarm Intell.* **2007**, *1*, 33–57. [\[CrossRef\]](#)
9. Kao, Y.-T.; Zahara, E. A hybrid genetic algorithm and particle swarm optimization for multimodal functions. *Appl. Soft Comput.* **2008**, *8*, 849–857. [\[CrossRef\]](#)
10. Juang, C.-F. A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. *IEEE Trans. Syst. Man. Cybern. Part B Cybern.* **2004**, *34*, 997–1006. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Garg, H. A hybrid PSO-GA algorithm for constrained optimization problems. *Appl. Math. Comput.* **2016**, *274*, 292–305. [\[CrossRef\]](#)
12. Farhang, Y. Development of the meta-heuristic of PSOGA with K-means algorithm. *Int. J. Comput. Sci. Netw. Secur.* **2017**, *17*, 29–36.
13. Akbulut, U.; Cifci, M.A.; Aslan, Z. Hybrid modeling for stream flow estimation: Integrating machine learning and federated learning. *Appl. Sci.* **2023**, *13*, 10203. [\[CrossRef\]](#)
14. Akbulut, U.; Aslan, Z. Optimization of the use and planning of water resources with advanced data processing methods. *Aksaray Univ. J. Sci. Eng.* **2022**, *6*, 79–94. [\[CrossRef\]](#)
15. Akbulut, U.; Çifçi, M.A.; İşler, B.; Aslan, Z. Comparison of different techniques in river flow prediction with machine learning methods. *Gazi Üniversitesi Mühendis.-Mimar. Fakültesi Derg.* **2024**, *40*, 467–486. [\[CrossRef\]](#)
16. Reddy, M.J.; Nagesh Kumar, D. Multi-objective particle swarm optimization for generating optimal trade-offs in reservoir operation. *Hydrol. Process.* **2007**, *21*, 2897–2909. [\[CrossRef\]](#)
17. Montalvo, I.; Izquierdo, J.; Schwarze, S.; Pérez-García, R. Multi-objective particle swarm optimization applied to water distribution systems design: An approach with human interaction. *Math. Comput. Model.* **2010**, *52*, 1219–1227. [\[CrossRef\]](#)
18. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [\[CrossRef\]](#)
19. Singh, M.; Ahmed, S. IoT based smart water management systems: A systematic review. *Mater. Today Proc.* **2021**, *46*, 5211–5218. [\[CrossRef\]](#)
20. Coello Coello, C.A. Evolutionary multi-objective optimization: A historical view of the field. *IEEE Comput. Intell. Mag.* **2006**, *1*, 28–36. [\[CrossRef\]](#)
21. Herabad, M.G.; Taheri, J.; Ahmed, B.S.; Curescu, C. E-PSOGA: An enhanced hybrid metaheuristic for optimal edge-to-cloud placement of services with multi-version components. *IEEE Access* **2025**, *13*, 151170–151188. [\[CrossRef\]](#)
22. Hosseini, V.; Farhang, Y.; Majidzadeh, K.; Ghobadi, C. Customized mutated PSO algorithm of isolation enhancement for printed MIMO antenna with ISM band applications. *AEU-Int. J. Electron. Commun.* **2022**, *145*, 154067. [\[CrossRef\]](#)
23. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [\[CrossRef\]](#)
24. Trivedi, M.; Shrivastava, R.K. Reservoir operation management using a new hybrid algorithm of Invasive Weed Optimization and Cuckoo Search Algorithm. *AQUA Water Infrastruct. Ecosyst. Soc.* **2023**, *72*, 1607–1628. [\[CrossRef\]](#)
25. Le, X.; Ling, K.; Zhou, L.; Wen, Y. A novel hybrid biological optimisation algorithm for tackling reservoir optimal operation problem. *Ain Shams Eng. J.* **2025**, *16*, 103342. [\[CrossRef\]](#)

26. Labadie, J.W. Optimal operation of multireservoir systems: State-of-the-art review. *J. Water Resour. Plan. Manag.* **2004**, *130*, 93–111. [[CrossRef](#)]
27. Akbulut, U.; Cifci, M.A.; Özyavaş, A.; Aslan, Z.; Koçak, K. Hybrid Wavelet and Machine Learning Models for Monthly Streamflow Prediction in the Karamandere Creek Basin. Available online: <https://ssrn.com/abstract=6462629> (accessed on 31 July 2026).
28. Salari, M.; Foong, L.K.; Shahid, E.S.; Moayedi, H.; Cifci, M.A. Hybrid multilayer perceptron models optimized by evolutionary algorithms for urban air quality forecasting: A case study of Shiraz, Iran. *Sci. Rep.* **2026**, *16*, 21838. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.