water

MDPI

Article

Parameters Tuning of Fractional-Order Proportional Integral Derivative in Water Turbine Governing System Using an Effective SDO with Enhanced Fitness-Distance Balance and Adaptive Local Search

Weiguo Zhao [1], Hongfei Zhang [1], Zhenxing Zhang [2], Kaidi Zhang [1] and Liying Wang [1,*]

[1] School of Water Conservancy and Hydropower, Hebei University of Engineering, Handan 056038, China
[2] Illinois State Water Survey, Prairie Research Institute, University of Illinois at Urbana-Champaign, Champaign, IL 61820, USA
* Correspondence: wangliying@hebeu.edu.cn

Abstract: Supply-demand-based optimization (SDO) is a swarm-based optimizer. However, it suffers from several drawbacks, such as lack of solution diversity and low convergence accuracy and search efficiency. To overcome them, an effective supply-demand-based optimization (ESDO) is proposed in this study. First, an enhanced fitness-distance balance (EFDB) and the Levy flight are introduced into the original version to avoid premature convergence and improve solution diversity; second, a mutation mechanism is integrated into the algorithm to improve search efficiency; finally, an adaptive local search strategy (ALS) is incorporated into the algorithm to enhance the convergence accuracy. The effect of the proposed method is verified based on the comparison of ESDO with several well-regarded algorithms using 23 benchmark functions. In addition, the ESDO algorithm is applied to tune the parameters of the fractional-order proportional integral derivative (FOPID) controller of the water turbine governor system. The comparative results reveal that ESDO is competitive and superior for solving real-world problems.

Keywords: supply-demand-based optimization; enhanced fitness-distance balance; adaptive local search; Levy flight; FOPID; water turbine

Citation: Zhao, W.; Zhang, H.; Zhang, Z.; Zhang, K.; Wang, L. Parameters Tuning of Fractional-Order Proportional Integral Derivative in Water Turbine Governing System Using an Effective SDO with Enhanced Fitness-Distance Balance and Adaptive Local Search. *Water* **2022**, *14*, 3035. https://doi.org/10.3390/w14193035

Academic Editor: Chin H Wu

Received: 2 August 2022
Accepted: 23 September 2022
Published: 27 September 2022

1. Introduction

With the development of social economy and technology, many complex optimization problems have appeared in the fields of communication, transportation, machinery, e-commerce, automation, materials and economics [1–10]. Meta-heuristic algorithms, an effective tool, simulate one or some natural processes in nature and have unique advantages in solving complex optimization problems. The No-Free−Lunch (NFL) theorem [11] proves that any optimizers cannot provide the best solutions for all different optimization problems. Therefore, plenty of meta-heuristics have sprung up in recent years according to different inspirations. Meta-heuristic algorithms can be divided into three categories: evolutionary-based (EB), physis-based (PB), and swarm-based (SB).

The most representative EB is the genetic algorithm (GA) [12], which simulates the evolution process of biological groups in nature. With the development of GA, many improved versions and variants have emerged, but most of them obtain high-quality solutions through mutation, crossover and selection steps. Other EBs are genetic programming (GP) [13], evolution strategies (ES) [14] and evolutionary programming (EP) [15].

PBs simulate the physical laws in nature. The annealing algorithm (SA) is a classical PB. The theoretical idea behind SA is the motion of molecules in a solid material as it cools gradually from high temperature. In recent years, numerous new physics-inspired algorithms have been proposed, including gravitational search algorithm (GSA) [16], atom

search algorithm (ASO) [17], electromagnetic mechanism-like algorithm (EM) [18], central force optimization (CFO) [19], hysteretic optimization (HO) [20], charged system search (CSA) [21], Lichtenberg algorithm (LA) [22], henry gas solubility optimizer (HGSO) [23], optics inspired optimization (OIO) [24], and so on.

SBs simulate the collective behaviors of natural or artificially formed organizations. Particle swarm algorithm (PSO) [25], a well-known optimizer, has received a high degree of attention and widespread concern from scholars of many various fields. Since its emergence, other new SBs have been developed, including bacteria foraging optimization (BFO) [26], manta ray foraging optimization (MRFO) [27], virus colony search (VCS) [28], cuckoo search algorithm (CS) [29], artificial ecosystem-based optimization (AEO) [30], trees social relations optimization (TSR) [31], tuna swarm optimization (TSO) [32], black widow optimization algorithm (BWOA) [33], etc. Among three types of meta-heuristic algorithms, since SBs have an inherent advantage in exploration and exploitation, they have a larger family and are more competitive in solving complex optimization problems.

Inspired by supply-demand relation of commodities in economics, a swarm-based optimizer called supply-demand-based optimization (SDO) is proposed [34]. It simulates the cobweb model in market economy mathematically. Due to its simplicity and feasibility, it can achieve a balance between exploration and exploitation. Therefore, since the emergence of SDO, it has attracted a wide attention and has been applied in different fields. Ali et al. [35] used SDO for dielectric strength test and obtained the optimal value of SIR/TiO2 filler parameter percentage under different conditions. Ginidi et al. [36] extracted the electrical parameters of different photovoltaic models through SDO, and the comparisons with other algorithms showing that SDO is highly competitive. Alturki et al. [37] applied SDO to the selection problem of hybrid energy system, it can obtain lower loss of power supply probability (LPSP) and lower annualized system cost (ASC). Jing et al. [38] proposed an improved SDO algorithm for the prediction of foundation pit deformation, and the results demonstrated its feasibility and effectiveness. Ibrahim et al. [39] used SDO to estimate the parameters of induction motor and obtained a small deviation.

The operating state of the water turbine governor system is closely related to the stability of hydropower station operation and the safety of power consumption. Therefore, the parameter selection of the water turbine governor system becomes an important subject worthy of long-term study. At the beginning of the study, the parameters are obtained mainly through a large number of empirical formulas derived from field tests. However, the application value obtained by this method is not proportional to the time and financial consumption. The emergence of PID regulation law improves this situation, and is widely used because of its simple operation and good robustness. With the in-depth study of computer technology, it is found that traditional integer-order PID control law cannot be well modeled and analyzed for some complex non-linear engineering problems. The water turbine governor system has strong memory dependence, and its dynamic process can be better described by fractional-order calculus. The most intuitive difference between fractional-order PID (FOPID) controller and traditional integer-order PID controller is that there are two more adjustable parameters in FOPID, which can achieve better control effect. Therefore, the FOPID controller has strong application value.

The rise of intelligent optimization techniques provides a more effective method for parameter adjustment of FOPID controller. Vanchinathan et al. [40] applied the artificial bee colony algorithm to tuning for brushless DC motor. Oguzhan Karahan [41] used FOPID controller in core power control in molten salt reactors and adjusted the parameters by using the cuckoo algorithm. Munagala [42] performed robustness analysis on FOPID controller in an AVR system by using chaotic black widow algorithm. These studies demonstrated the superiority of FOPID in effectiveness of different control systems.

Although SDO as a swarm-based optimizer has better optimization ability than some other optimizers when handling optimization problems, it still has some big promotion space. SDO has the limitations of solution diversity, search efficiency and solution accuracy when tracking complex problems. Therefore, to overcome the drawbacks, this study

designs an effective SDO termed ESDO; it incorporates the EFDB and Levy flight, mutation mechanism and adaptive local search strategy into SDO to perform global search. The effect of ESDO is verified based on the comparative results of ESDO with several well-regarded algorithms on a set of benchmark functions and tuning FOPID controller of water turbine governor. The results verify the efficiency and superiority of ESDO. The major difference of our study and its competitors is that ESDO makes three different strategies to improve its overall optimization performance in terms of solution diversity, convergence accuracy and search efficiency.

The rest of this study is as follows. Section 2 gives the main structure of SDO and provides the proposed ESDO by combining several strategies. In Section 3, the experimental results on some functions are investigated to assess the effectivity of the proposed ESDO. Section 4 provides an application of ESDO in tuning the FOPID controller of a water turbine governor. Section 5 gives some conclusions of the study.

2. Effective Supply-Demand-Based Optimization (ESDO)

2.1. Supply-Demand-Based Optimization (SDO)

In SDO [34], there are d commodity prices and quantities as candidate solutions and possible candidate solutions, respectively. After evaluation, the one with better fitness is selected as the current candidate solution. The mathematical expressions for commodity price and quantity are:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^d \\ \vdots & \vdots & \vdots & \vdots \\ x_n^1 & x_n^2 & \cdots & x_n^d \end{bmatrix}, \tag{1}$$

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} y_1^1 & y_1^2 & \cdots & y_1^d \\ y_2^1 & y_2^2 & \cdots & y_2^d \\ \vdots & \vdots & \vdots & \vdots \\ y_n^1 & y_n^2 & \cdots & y_n^d \end{bmatrix}, \tag{2}$$

where x_i and y_i $(i = 1 \cdots n)$ represent the price vectors and quantity vectors in each market, and x_i^j and $y_i^j (i = 1 \cdots n; j = 1 \cdots d)$ represent the price and quantity of the jth commodity in the ith market, respectively. There are n markets in total.

Then, the fitness values of all prices and quantities are evaluated by the following functions:

$$Fx = [Fx_1 Fx_2 Fx_3 \cdots Fx_n]^T, \tag{3}$$

$$Fy = [Fy_1 Fy_2 Fy_3 \cdots Fy_n]^T, \tag{4}$$

where T represents transposition of the matrix.

The vector of the equilibrium price and quantity are represented by:

$$\begin{cases} M_i = \left| Fx_i - \frac{1}{n} \sum_{i=1}^{n} Fx_i \right| \\ P = M / \sum_{i=1}^{n} M_i \\ x_0 = \begin{cases} r_1 \cdot \frac{1}{n} \sum_{i=1}^{n} x_i & \mathrm{r} < 0.5 \\ x_k, k = R(P) & \mathrm{r} \geq 0.5 \end{cases} \end{cases}, \tag{5}$$

$$\begin{cases} N_i = \left| Fy_i - \frac{1}{n} \sum_{i=1}^{n} Fy_i \right| \\ Q = N / \sum_{i=1}^{n} N_i \\ y_0 = y_k, k = R(Q) \end{cases}, \tag{6}$$

where r and r_1 are the random numbers in $[0,1]$, $R(P)$ and $R(Q)$ are the Roulette Wheel Selection.

The expressions of demand function and supply function are given by, respectively:

$$x_i(t) + 1 = x_0 - \beta(y_i(t) + 1 - y_0), \tag{7}$$

$$y_i(t) + 1 = y_0 + \alpha(x_i(t) - x_0), \tag{8}$$

where $x_i(t)$ and $y_i(t)$ respectively represent the ith price and quantity of commodity at time t, and β and α are respectively the demand weight and supply weight.

Substituting (7) into (8), the demand equation can be rewritten into the following form:

$$x_i(t) + 1 = x_0 - \alpha\beta(x_i(t) - y_0), \tag{9}$$

Therefore, the commodity price is updated by adjusting the values of α and β, and it is updated according to the equilibrium price relative to the current price. α and β can be expressed as:

$$\alpha = \frac{2(T - t + 1)}{T} sin(2\pi r), \tag{10}$$

$$\beta = 2cos(2\pi r), \tag{11}$$

where, $|\alpha\beta| < 1$ is equivalent to the stability mode in the supply and demand mechanism, emphasizing the exploitation ability, $|\alpha\beta| > 1$ is equivalent to the instability mode and emphasizing the overall exploration ability. Figure 1 shows the two modes of SDO.

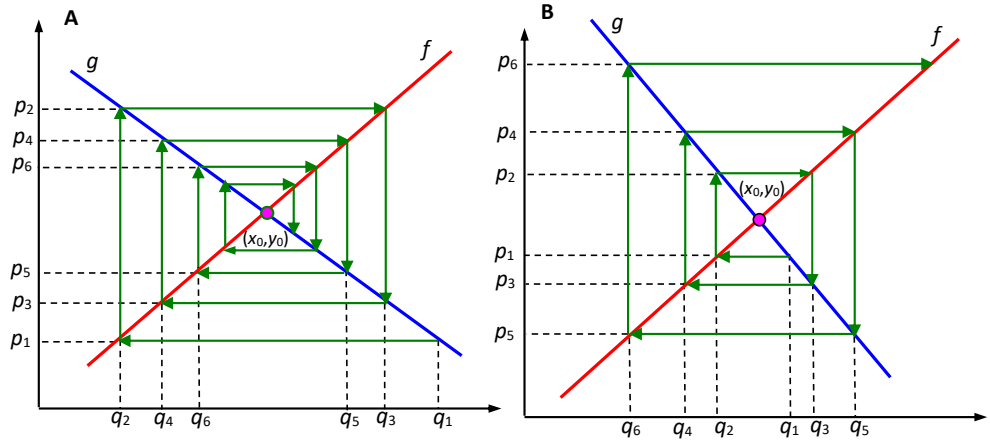

Figure 1. (**A**) stability mode, (**B**) instability mode [34].

2.2. Proposed Method

Since SDO easily suffers from low search efficiency and misses some better solutions, ESDO is an enhanced version of SDO in order to overcome them. For SDO, the fitness value of each solution and the distance between each solution and the best solution are very two important factors affecting the search efficiency of the algorithm for the optimal solution. In this study, inspired by a fitness-distance balance (FDB) [43], an enhanced fitness-distance balance (EFDB) is developed to replace the selection for the equilibrium quantity and price of commodities. Meanwhile, to strengthen exploration ability of the algorithm in the search space, a Levy flight strategy is introduced to the weight to improve the convergence ability. A mutation mechanism is employed to enhance the search efficiency of SDO. Meanwhile, an adaptive local search strategy is used to improve the convergence accuracy of the algorithm. ESDO is a newly presented optimizer and not yet applied in any real-world application.

2.2.1. Combining Enhanced Fitness-Distance Balance (EFDB) and Levy Flight

In [44], Kati et al. proposed an improved version of SDO, in which the equilibrium quantity is replaced with the commodity quantity selected by the FDB method to provide

diversity. In this study, according to [44], we propose the EFDB method, in which the equilibrium quantity and price of commodities are replaced with the commodity quantity and price by using the FDB method, respectively. This method can further strengthen the solution diversity. The following is a specific description to the EFDB method. The fitness value of commodity price is calculated by [43]:

$$\forall_{i=1}^{N} Fx_i = \begin{cases} \text{if goal is minimization } 1-normG_i \\ \text{if goal is maximization } normG_i \end{cases} \tag{12}$$

$$Fx = [fx_1, \cdots, fx_N]1_{\times N} \tag{13}$$

$$\forall_{i=1}^{n}, i \neq best, D_i = \sqrt{\left(x_i^1 - x_{best}^1\right)^2 + \cdots + \left(x_i^d - x_{best}^d\right)^2} \tag{14}$$

$$Dx = [dx_1, \cdots, dx_n]1_{\times n} \tag{15}$$

$$\forall_{i=1}^{n}, M_i = 0.5(1 + It/MaxIt) \times normFx_i + 0.5(1 - It/MaxIt) \times normDx_i \tag{16}$$

The equilibrium price vector x_0 is redefined by:

$$P = \frac{M_i}{\sum_{i=1}^{n} M_i} \tag{17}$$

As an analogy, the equilibrium quantity vector y_0 is redefined by:

$$Q = \frac{N_i}{\sum_{i=1}^{n} N_i} \tag{18}$$

where

$$\forall_{i=1}^{n}, M_i = 0.5(1 + It/MaxIt) \times normFy_i + 0.5(1 - It/MaxIt) \times normDy_i \tag{19}$$

where $normFy_i$ is the normalized fitness value of Fy_i, $normDy_i$ is the normalized distance value of the ith commodity, It is the current iteration and $MaxIt$ is the maximum number of iterations. This selection method combines the fitness and the distance to calculate the score for each individual in the population. Therefore, this strategy can effectively improve solution diversity and avoid local solutions. In addition, inspecting Equation (16), in the early iterations, the selection for the equilibrium quantity and price of commodities takes into account the largest fitness value and the furthest distance from the current optimal individual so far, it will contribute to exploration; in the later iterations, the selection focuses more on the fitness value rather than the distance, it will be dedicated to exploitation.

The Levy flight [45,46] whose step-width obeys non-uniform Levy distribution, is a random walk; thus, it has the superior ability to enhance exploring space search [47]. The step-width of the Levy flight is produced by [48,49]:

$$L(s) \sim t^{-\lambda} \ (1 \leq \lambda \leq 3) \tag{20}$$

$$s = \frac{u\sigma_u}{|v|^{\frac{1}{b}}} \tag{21}$$

where λ is a stability/tail index, s is the step-width and u and v obey the normal distribution, respectively:

$$u \sim N(0,1), v \sim N(0,1), \tag{22}$$

$$\sigma_u = \left(\frac{\Gamma(1+b) \cdot \sin(\frac{\pi b}{2})}{\Gamma(\frac{1+b}{2}) \cdot \beta \cdot 2^{\frac{b-1}{2}}} \right)^{\frac{1}{b}} \tag{23}$$

where Γ denotes the standard Gamma function and b = 1.5.

The weight α in Equation (10) is reformulated by:

$$\alpha = \frac{\frac{5u\sigma_u}{2|v|^{\frac{1}{\beta}}} \cdot (T - t + 1)}{T} \cdot \sin(2\pi r) \tag{24}$$

2.2.2. The Mutation Mechanism

To improve the search efficiency of the algorithm, the mutation mechanism is employed in this study. Although some different mutation strategies are introduced in the literature [50–52], the Gaussian mutation is one of the most frequently used mutation methods since it is more effective and simpler to implement [53]. So, in ESDO, the supply function and demand function are modified by, respectively:

$$y_i(t+1) = y_0 + \alpha \cdot (x_i(t) - x_0) + round(0.5 \cdot (0.01 + rand)) \cdot rn \tag{25}$$

$$x_i(t+1) = x_0 - \beta \cdot (y_i(t+1) - y_0) + round(0.5 \cdot (0.01 + rand)) \cdot rn \tag{26}$$

where

$$rn \sim N(0,1) \tag{27}$$

2.2.3. Adaptive Local Search (ALS) Strategy

Local search strategy is an important way to improve the current best solution. The chaotic local search is a classic local search (CLS) method [54], which used the chaotic map to improve the solution quality by searching the neighborhood around the best solution so far. However, the step-size for this local search cannot decrease as iterations go on, which will affect the solution accuracy and search efficiency. So, to dynamically adjust the step-size of the local search, an adaptive local search (ALS) strategy is given by:

$$x'_{best}(t) = x_{\text{best}}(t) + step \cdot (2 \cdot rand - 1) \cdot (Ub - Lb) \tag{28}$$

$$step = e^{-\frac{20t}{T}} \tag{29}$$

$x'_{best}(t)$ is a new current solution generated at time t.

The update of the current best solution is given by:

$$\begin{cases} x_{best}(t+1) = x'_{best}(t) \text{ if } fit(x'_{best}(t)) < fit(x_{best}(t)) \\ x_{best}(t+1) = x_{\text{best}}(t) \text{ else} \end{cases}, \tag{30}$$

If the fitness value of $x'_{best}(t)$ is better than that of $x_{best}(t)$, the current best solution is replaced with the new one, or it remains unchanged. It can be observed from Equation (28) that a bigger step-size contributes to exploration in the early iterations, with the increase of iterations and, a small step-size is greatly dedicated to exploitation. In addition to improving the convergence accuracy, this adaptive search strategy also strengthens the balance between exploration and exploitation to some extent.

2.2.4. The Proposed ESDO Algorithm

By introducing the EFDB method and Levy flight, the mutation mechanism, and the ALS strategy to strengthen the optimization performance, the ESDO algorithm is proposed. The pseudocode of ESDO algorithm is given in Figure 2.

Initialize the market population and weights: the commodity price vectors $\boldsymbol{x_i}$ and commodity quantity vectors $\boldsymbol{y_i}$ are randomly initialized, calculate their fitness values Fx_i and Fy_i, replace $\boldsymbol{x_i}$ by $\boldsymbol{y_i}$ if Fy_i is better Fx_i, and $\boldsymbol{x}_{best}$= the best solution found so far. The weights α and β are set.

While the stop criterion is not satisfied do

For each market (i=1,...,n)

Determine the equilibrium quantity $\boldsymbol{y_0}$ by Eqs. (18)–(19) and the equilibrium price $\boldsymbol{x_0}$ by Eqs. (16)–(17), respectively.

Perform Levy flight by Eqs. (20)–(24)

Update the commodity quantity vector $\boldsymbol{y_i}$ by Eq. (25).

Update the commodity price vector $\boldsymbol{x_i}$ by Eq. (26).

Calculate their fitness values Fx_i and Fy_i.

If Fy_i is better than Fx_i,

replace $\boldsymbol{x_i}$ by $\boldsymbol{y_i}$,

End If.

End For.

Perform adaptive local search strategy by Eqs. (28)–(30)

Update the best solution found so far $\boldsymbol{x}_{best}$.

End While.

Return the best solution found so far $\boldsymbol{x}_{best}$.

Figure 2. Pseudocode of ESDO algorithm.

3. Experimental Results and Analysis

3.1. Test Functions and Parameter Setting

To assess the performance of the ESDO algorithm, a classical suit of benchmark set, containing 23 test functions (see Table A1 in Appendix A for details), are employed. The 23 benchmark functions include 7 unimodal functions (UFs) (F1–F7), 6 multimodal functions (MFs) (F8–F13) and 10 low-dimensional multimodal functions (LMFs) (F14–F23). Meanwhile, several competitive optimizers, including whale optimization algorithm (WOA) [55], gray wolf optimizer (GWO) [56], and GSA, are used and their results are provided for a comparison. For all the considered optimizers, the population size and the maximum number of iterations are set to 50 and 500, respectively. The ESDO algorithm is firstly analyzed qualitatively based on exploration and exploitation. Then, the Wilcoxon signed-rank test and Friedman test are statistically analyzed, respectively. The experimental results are based on 30 independent runs. The other parameter settings of all the considered optimizers are described in Table 1.

Table 1. Parameter setting of each algorithm.

Algorithm	Parameters	Values
GSA	Gravitational constant; decreasing coefficient	100; 20
WOA	Control parameter	[0, 2]
SDO	Convergence factor	linearly decreases from 2 to 0
GWO	Control parameter	[0, 2]
ESDO	Convergence factor	linearly decreases from 2 to 0

3.2. Exploitation Analysis

The functions F1–F7 having only one extremum are used to assess exploitation of algorithms. The comparison results of the algorithms on these UNs are listed in Table 2, in which the underline indicates the best value among all the algorithms. From this table, ESDO provides better solutions on functions F1–F5 and F7 in terms of mean and Std. ESDO performs as well as SDO and GWO on function F6. Therefore, ESDO obtains better results on most of UNs. Figure 3 shows the convergence curves of the algorithms on F1–F7. ESDO exhibits superior convergence rate over other optimizers in exploiting the optimal solution.

Table 2. Comparison results of UNs (F1–F7).

NO.	Index	ESDO	SDO	WOA	GWO	GSA
F1	Mean	$\underline{9.09 \times 10^{-231}}$	1.49×10^{-165}	2.62×10^{-83}	1.87×10^{-33}	3.66×10^{-17}
	Std	$\underline{0}$	0	9.62×10^{-83}	2.21×10^{-33}	1.15×10^{-17}
F2	Mean	$\underline{4.93 \times 10^{-116}}$	1.30×10^{-71}	2.40×10^{-53}	8.85×10^{-20}	3.21×10^{-8}
	Std	$\underline{2.70 \times 10^{-115}}$	7.04×10^{-71}	8.38×10^{-53}	5.55×10^{-20}	6.07×10^{-9}
F3	Mean	$\underline{3.86 \times 10^{-194}}$	3.66×10^{-128}	30167.4659	1.01×10^{-7}	5.38×10^{2}
	Std	$\underline{0.00 \times 10^{0}}$	1.99×10^{-127}	8562.673562	4.27×10^{-7}	2.49×10^{2}
F4	Mean	$\underline{3.28 \times 10^{-113}}$	3.22×10^{-75}	36.64494791	1.83×10^{-8}	3.31×10^{0}
	Std	$\underline{1.79 \times 10^{-113}}$	1.76×10^{-74}	29.45025131	1.45×10^{-8}	1.56×10^{0}
F5	Mean	$\underline{2.40 \times 10^{0}}$	2.59×10^{1}	2.75×10^{1}	2.64×10^{1}	3.85×10^{1}
	Std	$\underline{1.11 \times 10^{0}}$	7.19×10^{-1}	4.09×10^{-1}	6.35×10^{-1}	3.18×10^{1}
F6	Mean	$\underline{0}$	$\underline{0}$	3.33×10^{-2}	$\underline{0}$	6.33×10^{-1}
	Std	$\underline{0}$	$\underline{0}$	1.83×10^{-1}	$\underline{0}$	8.09×10^{-1}
F7	Mean	$\underline{7.49 \times 10^{-5}}$	9.85×10^{-5}	2.14×10^{-3}	1.14×10^{-3}	3.09×10^{-2}
	Std	$\underline{7.58 \times 10^{-5}}$	7.61×10^{-5}	2.91×10^{-3}	5.78×10^{-4}	1.73×10^{-2}

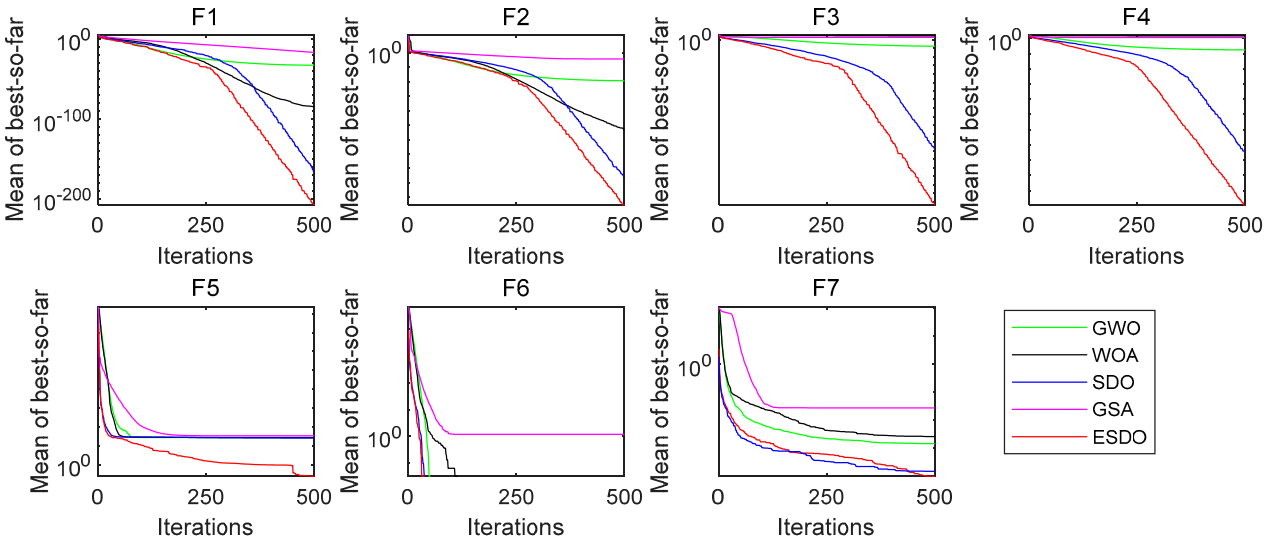

Figure 3. Convergence comparisons of UNs.

3.3. Exploration Analysis

The functions F8–F23 having multiple extrema are used to evaluate exploration of algorithms. The comparison results of the algorithms on these functions are shown in Tables 3 and 4. Inspecting Table 3, ESDO obviously outperforms other optimizers on function F12 and F13. ESDO, SDO and WOA provide the same results in terms of mean on functions F9 and F10. In addition, ESDO is third best algorithm on function F7. The performance of ESDO ranks only second to WOA on function F8. Observing Table 4, for

F15 and F20, ESDO obtains the second-best results, which are only inferior to those of SDO. For functions F16–F19 and F22–F23, ESDO, SDO and one of other algorithms offer the same best results. For F21, ESDO provides the best results. The convergence curves of the algorithms on F8–F23 are depicted in Figures 4 and 5, which manifests that ESDO has better convergence performance with comparison to other algorithms when tackling these test functions.

Table 3. Comparison results of MNs (F8–F13).

NO.	Index	ESDO	SDO	WOA	GWO	GSA
F8	Mean	−8525.069636	−8498.031439	−11,263.28556	−5998.877219	−2735.739471
	Std	5.93×10^{2}	7.29×10^{2}	1.55×10^{3}	1.19×10^{3}	3.67×10^{2}
F9	Mean	0	0	0	1.517856678	18.27407681
	Std	0	0	0	2.13×10^{0}	4.47×10^{0}
F10	Mean	8.88×10^{-16}	8.88×10^{-16}	5.27×10^{-15}	4.21×10^{-14}	5.05×10^{-9}
	Std	0	0	2.41×10^{-15}	4.53×10^{-15}	8.67×10^{-10}
F11	Mean	0	0	0	5.48×10^{-3}	1.76×10^{1}
	Std	0	0	0	9.90×10^{-3}	4.99×10^{0}
F12	Mean	5.77×10^{-5}	2.62×10^{-4}	5.68×10^{-3}	2.94×10^{-2}	6.31×10^{-1}
	Std	8.05×10^{-5}	4.09×10^{-4}	4.10×10^{-3}	1.75×10^{-2}	5.01×10^{-1}
F13	Mean	2.32×10^{-3}	1.93×10^{-2}	2.21×10^{-1}	3.40×10^{-1}	2.59×10^{0}
	Std	4.53×10^{-3}	3.96×10^{-2}	1.23×10^{-1}	1.97×10^{-1}	3.77×10^{0}

Table 4. Comparison results of LMNs (F14–F23).

NO.	Index	ESDO	SDO	WOA	GWO	GSA
F14	Mean	0.998003838	0.998003838	2.432300837	2.607801074	5.100884714
	Std	0	0	3.36×10^{0}	2.44×10^{0}	2.80×10^{0}
F15	Mean	3.07×10^{-4}	3.07×10^{-4}	7.54×10^{-4}	2.48×10^{-3}	3.28×10^{-3}
	Std	1.03×10^{-14}	1.30×10^{-17}	6.96×10^{-4}	6.07×10^{-3}	1.83×10^{-3}
F16	Mean	−1.031628453	−1.031628453	−1.031628453	−1.031628439	−1.031628453
	Std	6.78×10^{-16}	6.78×10^{-16}	1.05×10^{-10}	1.46×10^{-8}	5.05×10^{-16}
F17	Mean	0.397887358	0.397887358	0.397889341	0.397913346	0.397887358
	Std	0	0	4.92×10^{-6}	1.34×10^{-4}	0
F18	Mean	3	3	3.000019786	3.000012409	3
	Std	1.12×10^{-15}	1.18×10^{-15}	5.81×10^{-5}	1.44×10^{-5}	3.24×10^{-15}
F19	Mean	−3.862782148	−3.862782148	−3.859813445	−3.861222768	−3.862782148
	Std	2.70×10^{-15}	2.71×10^{-15}	3.69×10^{-3}	2.62×10^{-3}	2.31×10^{-15}
F20	Mean	−3.302179651	−3.310105859	−3.229898305	−3.259834789	−3.321995172
	Std	4.51×10^{-2}	3.63×10^{-2}	9.89×10^{-2}	7.99×10^{-2}	1.37×10^{-15}
F21	Mean	−10.15319968	−9.983266281	−9.389732298	−9.060734145	−7.353288325
	Std	7.01×10^{-15}	9.31×10^{-1}	2.01×10^{0}	2.26×10^{0}	3.54×10^{0}
F22	Mean	−10.40294057	−10.40294057	−8.736752324	−10.40158908	−10.40294057
	Std	1.04×10^{-16}	9.90×10^{-16}	2.85×10^{0}	6.32×10^{-4}	9.33×10^{-16}
F23	Mean	−10.53640982	−10.53640982	−7.471755875	−10.53516103	−10.53640982
	Std	1.21×10^{-15}	1.98×10^{-15}	3.59×10^{0}	5.47×10^{-4}	2.29×10^{-15}

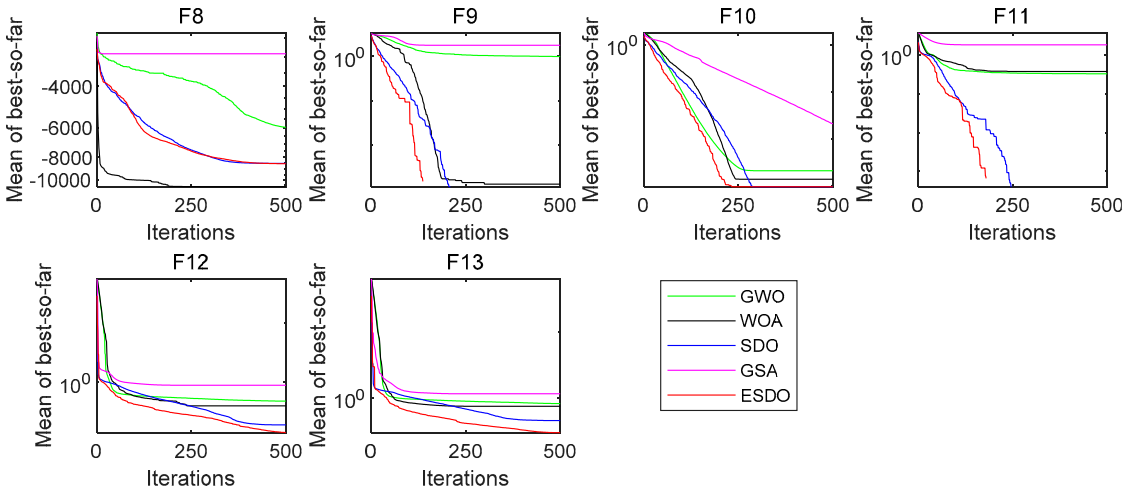

Figure 4. Convergence comparisons of MNs (F8–F13).

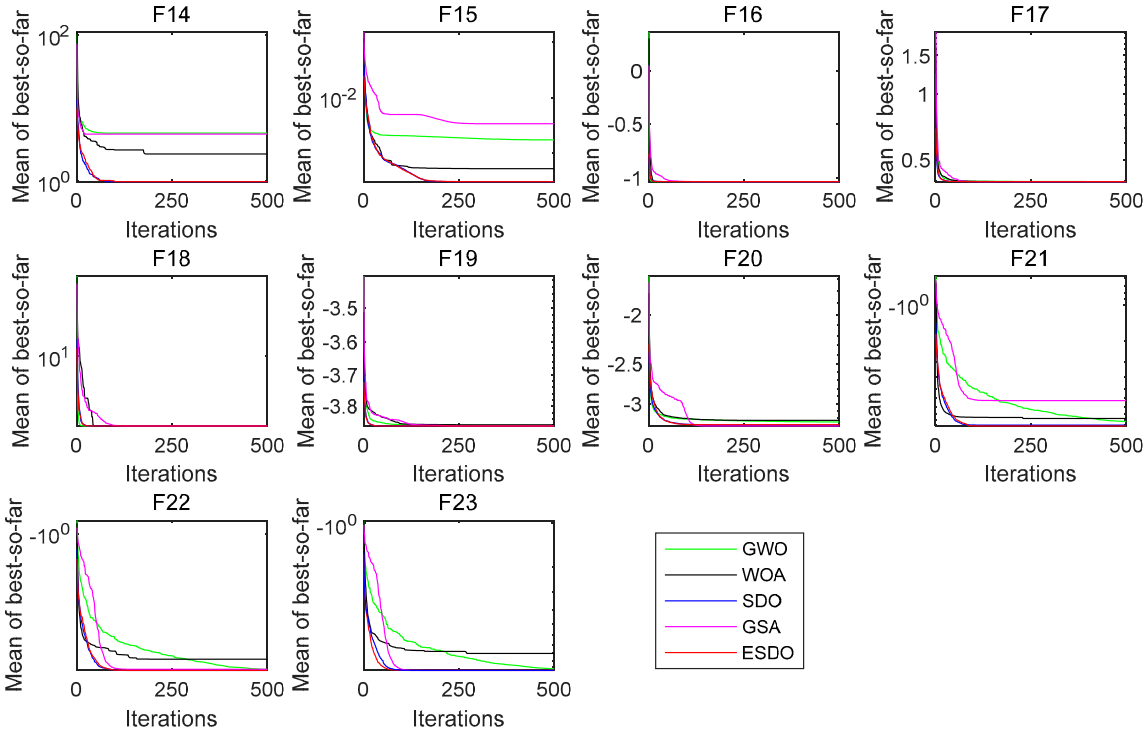

Figure 5. Convergence comparisons of LMNs (F14–F23).

3.4. Statistical Analysis

To evaluate the overall performance of ESDO and rank it statistically, the Wilcoxon signed-rank test (WSRT) [57] and Friedman test (FT) [58] are employed, respectively. Tables 5 and 6 provide the results of WSRT. In the table, "=" indicates there is no significant difference between ESDO and its competitor for a considered problem, "+" indicates that the performance of ESDO is better than that of a competitor for a considered problem and "−" is the opposite. Table 7 summarizes the results of WSRT in Tables 5 and 6. In Table 7, The WSRT results reveal that ESDO is superior to SDO, GWO, WOA and GSA on functions 7, 18, 22 and 17 out of 23, respectively, indicating the superior performance of ESDO to its competitors statistically. Figure 6 depicts the rank of each function for the comparative optimizers, Figure 7 gives the mean of these ranks. From Figure 7, ESDO ranks the first among the considered optimizers, demonstrating that it exhibits the best optimization ability compared to its counterparts.

Table 5. WSRT results of ESDO for SDO and WOA.

Fun.	SDO vs. ESDO				WOA vs. ESDO			
	p-Value	T+	T−	Winner	*p*-Value	T+	T−	Winner
F1	1.73×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F2	1.73×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F3	1.73×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F4	1.73×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F5	1.73×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F6	1	0	465	=	1	0	465	=
F7	3.68×10^{-2}	334	131	−	1.73×10^{-6}	0	465	+
F8	9.92×10^{-1}	233	232	=	6.89×10^{-5}	426	39	−
F9	1	0	465	=	1	0	465	=
F10	1	0	465	=	4.34×10^{-6}	0	465	+
F11	1	0	465	=	0.5	0	465	=
F12	4.49×10^{-2}	135	330	+	1.73×10^{-6}	0	465	+
F13	2.43×10^{-2}	123	342	+	1.73×10^{-6}	0	465	+
F14	1	0	465	=	1.73×10^{-6}	0	465	+
F15	5.73×10^{-1}	166	299	=	1.73×10^{-6}	0	465	+
F16	1	0	465	=	1.73×10^{-6}	0	465	+
F17	1	0	465	=	1.73×10^{-6}	0	465	+
F18	1	12	453	=	1.73×10^{-6}	0	465	+
F19	1	0	465	=	1.73×10^{-6}	0	465	+
F20	4.53×10^{-1}	19	446	=	4.86×10^{-5}	35	430	+
F21	1	0	465	=	1.73×10^{-6}	0	465	+
F22	1	0	465	=	1.73×10^{-6}	0	465	+
F23	1	0	465	=	1.73×10^{-6}	0	465	+

Table 6. WSRT results of ESDO for GWO and GSA.

Fun.	GWO vs. ESDO				GSA vs. ESDO			
	p-Value	T+	T−	Winner	*p*-Value	T+	T−	Winner
F1	1.73×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F2	1.73×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F3	1.73×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F4	1.73×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F5	1.73×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F6	1	0	465	=	1.56×10^{-2}	0	465	+
F7	1.73×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F8	1.73×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F9	4.63×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F10	1.31×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F11	3.91×10^{-3}	0	465	+	1.73×10^{-6}	0	465	+
F12	2.35×10^{-6}	3	462	+	1.73×10^{-6}	0	465	+
F13	1.92×10^{-6}	1	464	+	1.24×10^{-5}	20	445	+
F14	1.73×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F15	1.73×10^{-6}	0	465	+	1.73×10^{-6}	0	465	+
F16	1.73×10^{-6}	0	465	+	1	0	465	=
F17	1.73×10^{-6}	0	465	+	1	0	465	=
F18	1.73×10^{-6}	0	465	+	5.34×10^{-7}	15	450	+
F19	1.73×10^{-6}	0	465	+	1	0	465	=
F20	2.41×10^{-4}	54	411	+	6.25×10^{-2}	15	450	=
F21	1.73×10^{-6}	0	465	+	1.75×10^{-4}	0	465	+
F22	1.73×10^{-6}	0	465	+	1	0	465	=
F23	1.73×10^{-6}	0	465	+	1	0	465	=

Table 7. Statistics results of WSRT for ESDO.

Function Types	ESDO vs. SDO (+/=/−)	ESDO vs. WOA (+/=/−)	ESDO vs. WOA (+/=/−)	ESDO vs. GSA (+/=/−)
Unimodal	5/1/1	6/1/0	6/1/0	7/0/0
Multimodal	2/4/0	2/3/1	6/0/0	6/0/0
Low-dimensional	0/10/0	10/0/0	10/0/0	4/6/0
Total	7/15/1	18/4/1	22/1/0	17/6/0

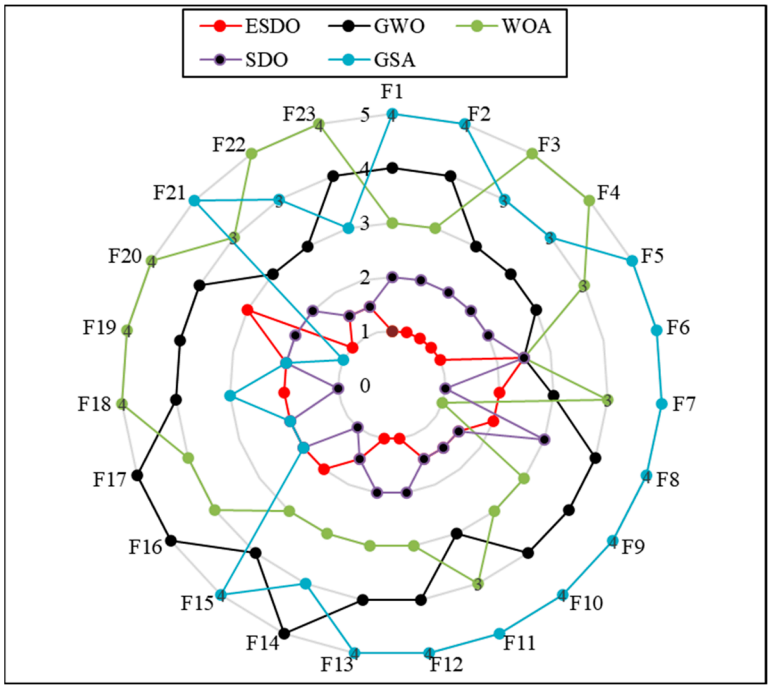

Figure 6. Ranks of different algorithms for each function.

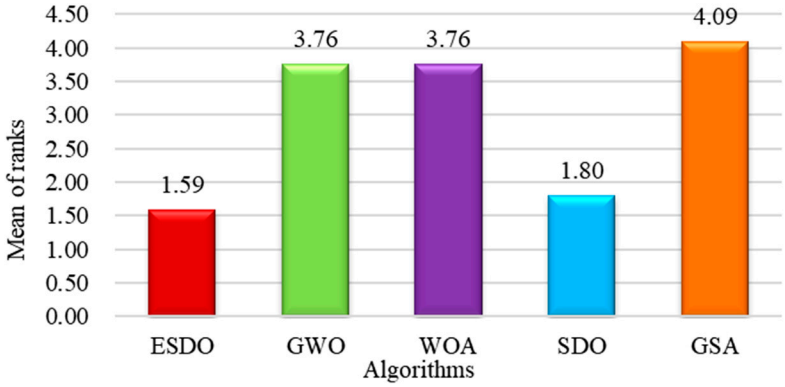

Figure 7. Mean of ranks of different algorithms.

4. ESDO for FOPID Controller of Water Turbine Governor System

4.1. System Description

The traditional integer-order PID controller has only three parameters (K_p,K_i,K_d) [59], while the FOPID controller used in this experiment has five parameters (K_p,K_i,K_d,λ,μ), so as to achieve more accurate control. Theoretically, λ and μ can take any number. In the case of $\lambda = \mu = 1$, FOPID is converted to integer-order PID.

The transfer function of FOPID is described by:

$$G_s(s) = K_p + \frac{K_i}{s^\lambda} + \frac{K_d \cdot s^\mu}{T_v \cdot s + 1}, \tag{31}$$

where K_p,K_i and K_d represent the proportional coefficient, integral coefficient and differential coefficient, respectively, s represents the Laplace operator, and λ and μ represent the exponents of the integral operator and the differential operator, respectively. T_v represents the differential time constant.

The variation of the unit load produces a deviation e. The FOPID controller will convert this difference into an adjustment signal. After receiving the signal, the mechanical hydraulic system will operate to adjust the opening of the guide vane, and then adjust the flow and restore the speed. Figure 8 is the FOPID model of the water turbine governor system.

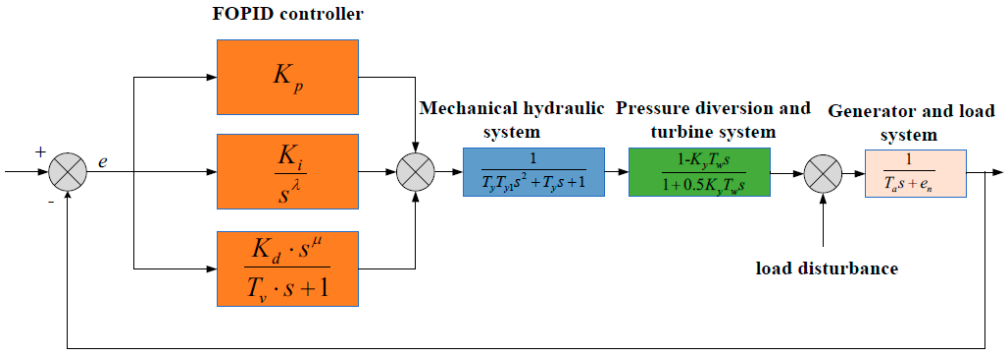

Figure 8. FOPID block diagram of the water turbine governor system.

4.2. Experimental Results and Analysis

This study is simulated by MATLAB/Simulink. The fitness function uses the time multiplication error integration criterion and is defined by:

$$f_{ITAE} = \int_0^{t_s} t|e(t)|dt, \tag{32}$$

where $e(t)$ is the deviation of the actual output from the expected output.

The specific experimental settings and parameter values are shown in Table 8.

Table 8. The specific experimental settings and parameter values.

Population Size	Maximum Numbers of Iterations	K_p	K_i	K_d	λ	μ
30	100	[0, 20]	[0, 20]	[0, 20]	[0.5, 1.9]	[0.5, 1.9]
T_v	T_y	T_{y1}	K_y	T_w	T_a	e_n
0.2	0.3	0.3	1	0.5	10	1

In this experiment, in order to be more practical, four different working conditions under 0–20% load are adopted, and the controller performance from ESDO is compared with that from SDO, WOA and GSA under each working condition. Each algorithm runs 20 times, and the average optimal fitness value, overshoot and adjustment time based on 20 experiments are compared to verify the parameter setting effect of ESDO on the FOPID controller. Under different load conditions, the overshoot is represented by the maximum value. The simulation results are shown in Table 9. The convergence curve of average fitness, convergence curves of average FOPID parameters and the speed response curves obtained of the average FOPID parameters under different load conditions are shown in Figures 9–20, respectively.

Table 9. Experimental results at 4%, 8%, 12%, and 16% loads.

Load	Average	Algorithms				
		ESDO	**SDO**	**GWO**	**WOA**	**GSA**
4%	*ITAE*	1.5028	1.7009	1.9237	4.0746	5.5928
	K_p	12.6487	12.0457	11.7392	10.3199	10.678
	K_i	4.5888	4.2521	4.5135	5.0279	4.0414
	K_d	3.3403	3.0282	3.3705	3.6921	3.3575
	λ	1.0228	1.0230	1.0153	0.9987	1.0830
	μ	1.5079	1.4519	1.4280	1.3416	1.3857
	Overshoot	0.4992	0.5100	0.5070	0.5113	0.5072
	T_s(s)	10.16	10.81	15.20	21.78	28.5100
8%	*ITAE*	3.1891	3.4378	4.3942	8.2490	19.0396
	K_p	11.4295	10.7352	10.2992	8.5996	10.2509
	K_i	4.2984	4.1532	4.1279	4.6332	4.2505
	K_d	3.39281	3.2958	3.1801	3.7471	3.7841
	λ	1.0234	1.0211	1.0156	0.9679	1.0563
	μ	1.4341	1.3563	1.3611	1.1568	1.3633
	Overshoot	1.0073	1.0274	1.0272	1.0601	1.0064
	T_s(s)	10.02	10.77	10.77	27.9500	28.7300
12%	*ITAE*	5.1116	5.2596	6.2520	12.2974	33.1630
	K_p	10.6533	10.2649	10.2900	7.7878	8.9427
	K_i	4.0361	3.8655	3.7692	4.4630	3.3708
	K_d	3.7796	3.1963	3.1225	3.9908	4.0885
	λ	1.0233	1.0229	1.0236	0.9653	1.0339
	μ	1.4056	1.3735	1.4244	1.1403	1.3098
	Overshoot	1.5073	1.5338	1.5135	1.5806	1.5024
	T_s(s)	10.32	10.45	10.42	22.58	28.09
16%	*ITAE*	8.4368	9.0978	9.9455	18.8710	40.3035
	K_p	10.1548	9.6269	9.2327	7.4238	9.2411
	K_i	3.4610	3.3166	3.3739	3.4810	3.1995
	K_d	3.7743	2.7425	3.2679	3.7848	3.9186
	λ	1.0243	1.0197	1.0170	0.9832	1.0734
	μ	1.3936	1.3859	1.3315	1.1247	1.3524
	Overshoot	2.0653	2.1087	2.1005	2.1538	2.0577
	T_s(s)	11.09	11.15	13.40	27.29	27.90

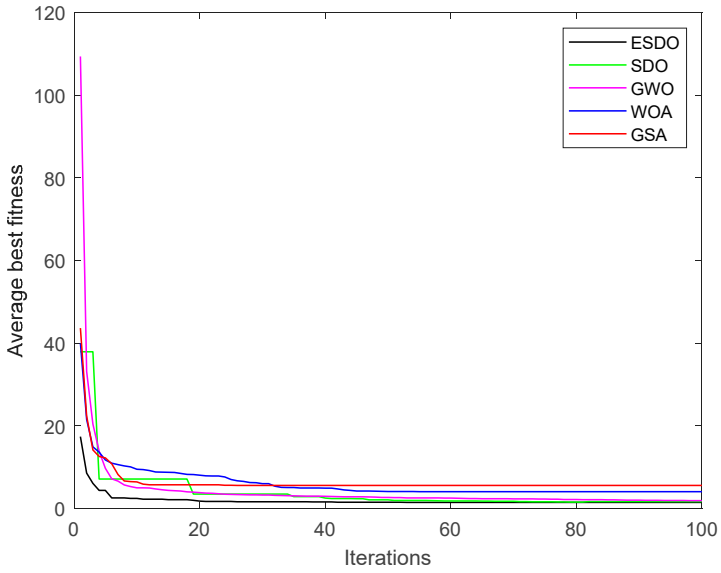

Figure 9. Convergence curves of average fitness with 4% load.

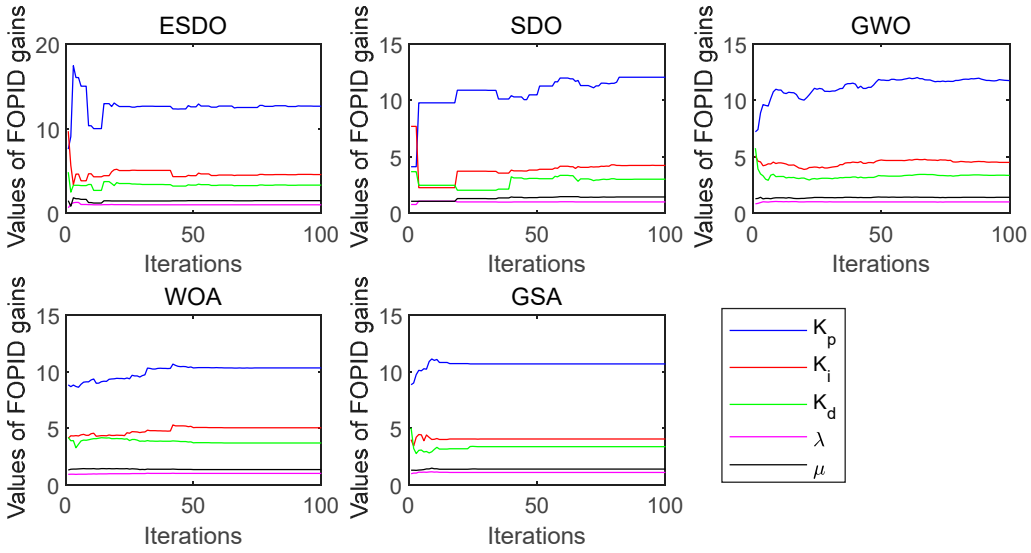

Figure 10. Convergence curves of FOPID parameters with 4% load.

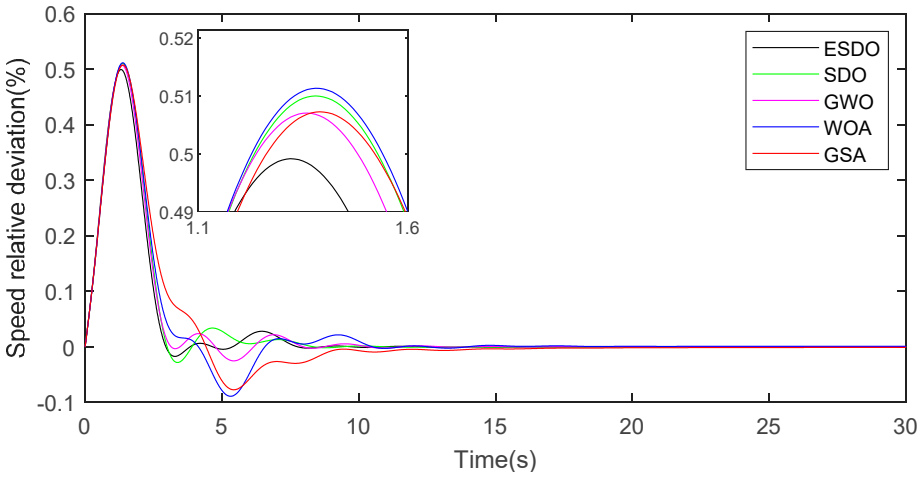

Figure 11. Speed response curves with 4% load.

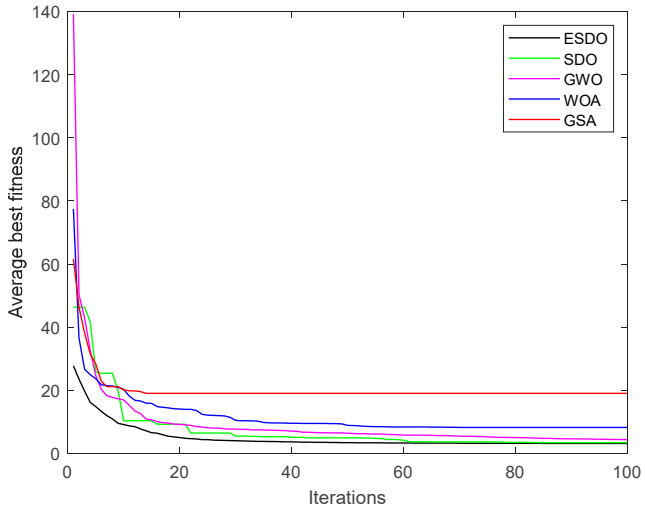

Figure 12. Convergence curves of average fitness with 8% load.

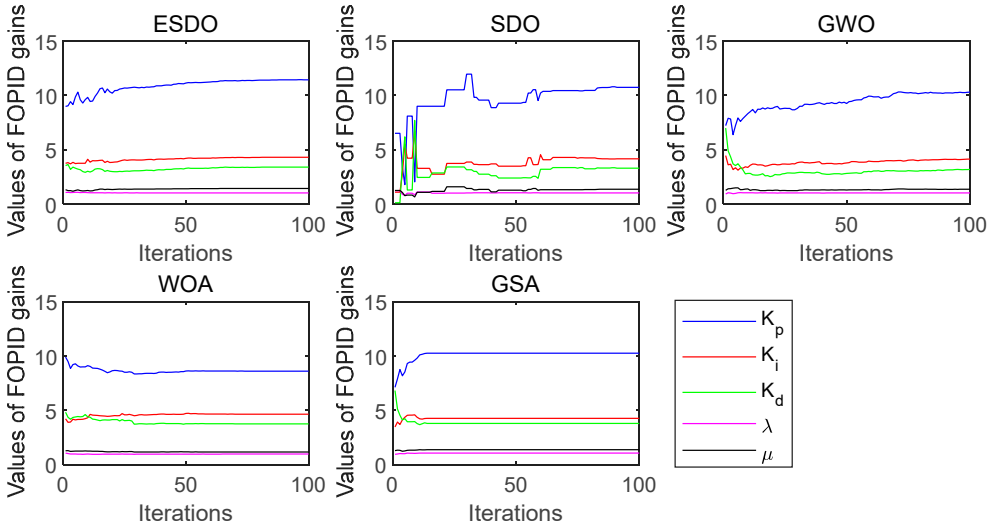

Figure 13. Convergence curves of FOPID parameters with 8% load.

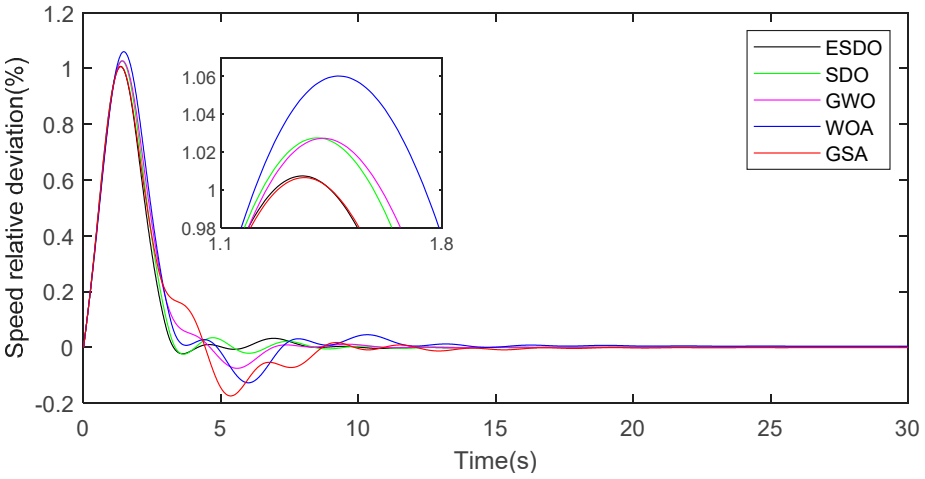

Figure 14. Speed response curves with 8% load.

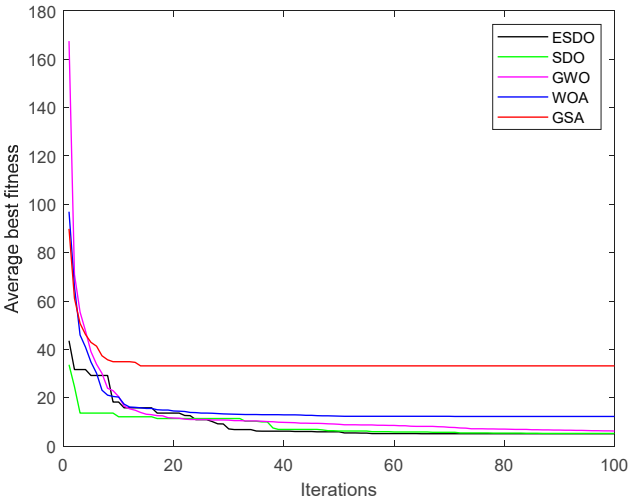

Figure 15. Convergence curves of average fitness with 12% load.

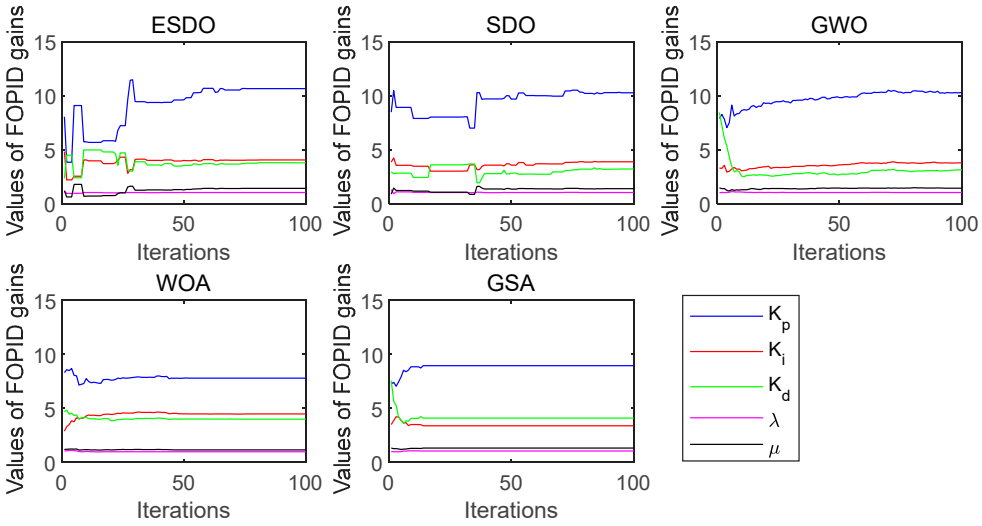

Figure 16. Convergence curves of FOPID parameters with 12% load.

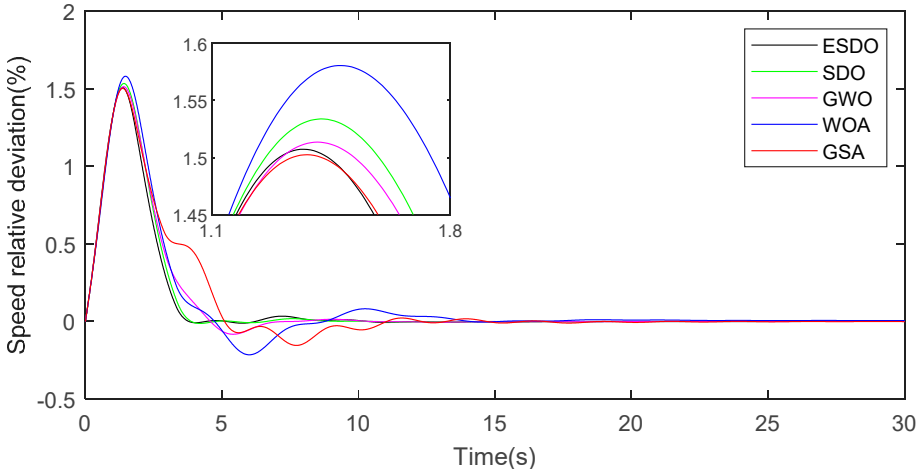

Figure 17. Speed response curves with 12% load.

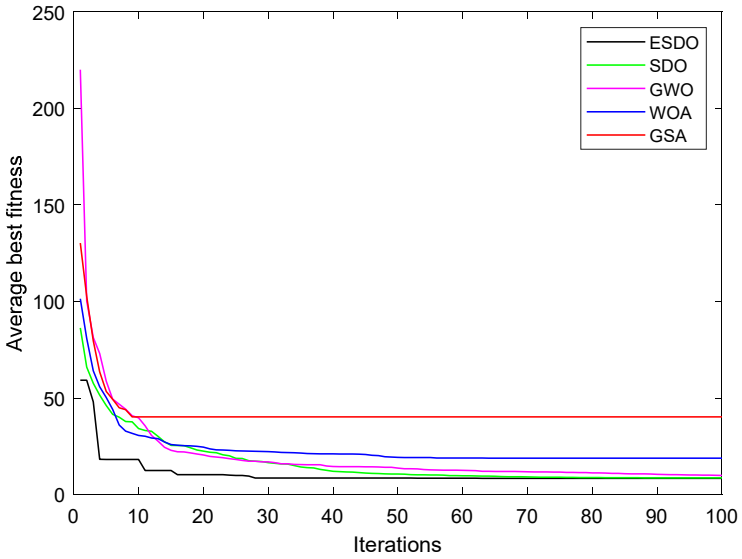

Figure 18. Convergence curves of average fitness with 16% load.

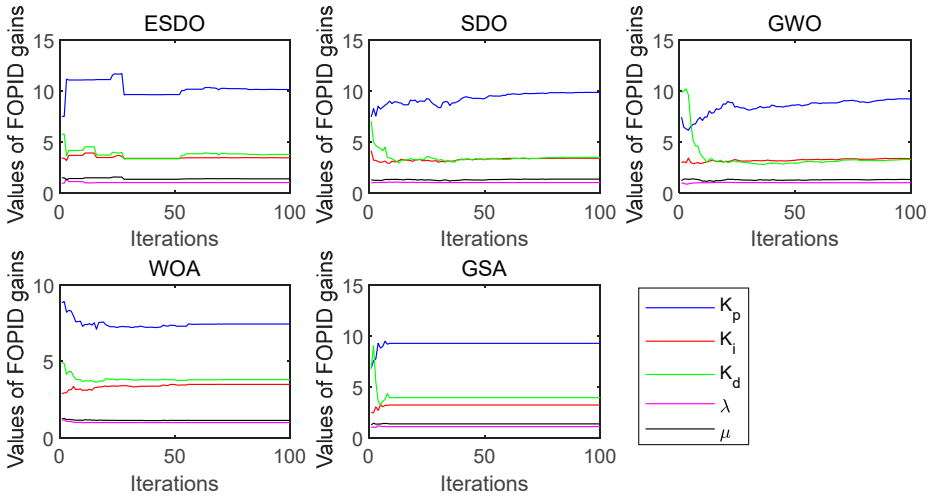

Figure 19. Convergence curves of FOPID parameters with 16% load.

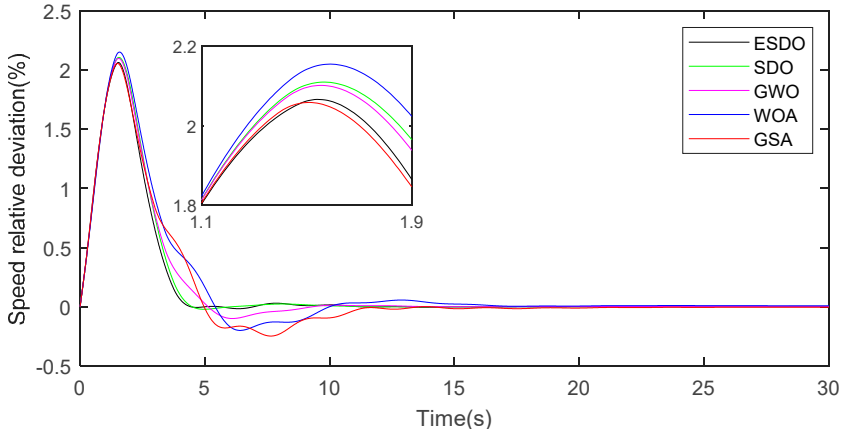

Figure 20. Speed response curves with 16% load.

Under 4% load, the fitness value of the FOPID controller offered by ESDO is the smallest among the five algorithms, which shows that ESDO can jump out of local optimum well and find the global optimum; meanwhile, it effectively improves the optimization accuracy. From the convergence curves under 4% load in Figure 9, the convergence rate of ESDO is also faster than other algorithms, which indicates that ESDO has the strongest ability to find the best solution among these algorithms. The overshoot and adjustment time are two important indicators to measure whether a control system is stable. It can be seen from Table 9 that under the condition of 4% load, the overshoot and adjustment time obtained are the minimum after the governor of the water turbine is tuned by ESDO. The trend of the response curves in Figure 11 shows that after the peak value, the oscillation amplitude of the curve obtained by ESDO is the smallest, and it can reach a stable value quickly. On the contrary, the oscillation amplitude of the response curves obtained by SDO, GWO, WOA and GSA are very large, and the state is very unstable. It proves that the dynamic regulation ability of the FOPID controller offered by ESDO is stronger than those offered by the other four algorithms, and the turning parameters of the turbine governor with ESDO can make the control system more stable and efficient.

Under 8% load, the fitness value of the FOPID controller offered by ESDO is also the smallest, showing that the global exploration ability and local exploitation ability of ESDO are significantly enhanced. The convergence curves of average fitness in Figure 12 demonstrate that, under the 8% load condition, ESDO can quickly find the optimal value in the optimization process, greatly improving the efficiency of the algorithm. After the governor of the water turbine is tuned by different algorithms, the overshoot of the FOPID

controller offered by ESDO is only second to that offered by GSA and very close to that offered by GSA, but the adjustment time of the FOPID controller offered by ESDO is far better than that offered by GSA. It can be found from Figure 14 that the FOPID controller from ESDO also has the smallest oscillation amplitude and the smallest duration, the FOPID controller from GSA has the largest oscillation amplitude and the longest duration, and the FOPID controllers from SDO, GWO and WOA are in between. Considering comprehensively, under 8% load condition, ESDO also has the strongest optimization ability for the FOPID controller among the five algorithms.

When the load is 12%, the FOPID controller offered by ESDO also gets the minimum fitness value. From Figure 15, although the convergence speed of ESDO is slower than that of SDO at the beginning of the iteration, in the middle of the iteration, ESDO outperforms SDO. At this time, the convergence speed of ESDO is the fastest, and it tends to jump out of the local extreme value to find a smaller value. Moreover, in the regulation system, the overshoot of the FOPID controller offered by ESDO is relatively good, and the adjustment time is also the smallest. From the speed response curves in Figure 17, after the first peak, the ESDO curve is very stable with little fluctuation, while the response curves of the FOPID controller offered by other algorithms fluctuate greatly and last a long time.

When the load is 16%, the fitness value of the FOPID controller offered by ESDO is still the smallest. From Figure 18, there is a fast convergence speed at the beginning of iteration, and its ability to jump out of local extreme value is also the strongest among all algorithms. According to the speed response curves in Figure 20, the overshoot of of the FOPID controller offered by ESDO is very close to that offered by GSA, which is better than those offered by other algorithms. The adjustment time of the FOPID controller offered by ESDO is the shortest among all algorithms, after falling from the peak, the fluctuation weakens rapidly and finally tends to be stable.

Therefore, through tuning the FOPID parameters of the water turbine governor under different load conditions, these convincing results reveal that ESDO is excellent in tackling real-world engineering applications.

5. Conclusions

To better handle optimization problems, an effective supply-demand-based optimization (ESDO) is proposed, it combines three strategies, including the enhanced fitness-distance balance (EFDB) with Levy flight, mutation mechanism and adaptive local search, to improve solution diversity, and convergence accuracy and search efficiency. The experimental results of ESDO with several well-regarded algorithms on 23 benchmark functions discover that ESDO has superior optimization ability. In addition, the practicability of ESDO is also verified by tuning the parameters of the FOPID controller of the water turbine governor system. The experimental results show that the turbine governor system tuned by ESDO is better than those tuned by other algorithms in terms of response time and overshoot.

In this study, some secondary and high-level terms are ignored when establishing the hydraulic turbine simulation model, and higher-order models can be established in the future. The implementation method of FOPID can also be improved.

In future studies, ESDO could be specifically intended to handle a variety of problems in hydraulic engineering, such as optimal allocation of power station [60], intelligent fault diagnosis of turbine [61] and optimal design of gate [62–64].

Author Contributions: Conceptualization, W.Z. and L.W.; methodology, H.Z. and W.Z.; software, Z.Z.; validation, Z.Z., K.Z. and H.Z.; formal analysis, H.Z.; investigation, writing—original draft preparation, W.Z and H.Z.; writing—review and editing, H.Z.; visualization, L.W.; supervision, L.W.; project administration L.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Appendix A

Table A1. Unimodal test functions.

Name	Function	D	Range	f_{opt}
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100, 100]^D$	0
Schwefel2.22	$f_2(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	30	$[-10, 10]^D$	0
Schwefel1.2	$f_3(x) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	30	$[-100, 100]^D$	0
Schwefel2.21	$f_4(x) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	30	$[-100, 100]^D$	0
Rosenbrock	$f_5(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i)^2 + (x_i - 1)^2)$	30	$[-30, 30]^D$	0
Step	$f_6(x) = \sum_{i=1}^{n} (x_i + 0.5)^2$	30	$[-100, 100]^D$	0
Quartic	$f_7(x) = \sum_{i=1}^{n} ix^4 + random[0, 1)$	30	$[-1.28, 1.28]^D$	0

Table A2. Multimodal test functions.

Name	Function	D	Range	f_{opt}
Schwefel	$f_8(x) = \sum_{i=1}^{n} (x_i \sin(\sqrt{\lvert x_i \rvert}))$	30	$[-500, 500]^D$	$-12,569.5$
Rastrigin	$f_9(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)^2$	30	$[-5.12, 5.12]^D$	0
Ackley	$f_{10}(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n} \cos 2\pi x_i) + 20 + e$	30	$[-32, 32]^D$	0
Griewank	$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} (x_i - 100)^2 - \prod_{i=1}^{n} \cos(\frac{x_i - 100}{\sqrt{i}}) + 1$	30	$[-600, 600]^D$	0
Penalized	$f_{12}(x) = \frac{\pi}{n}\{10\sin^2(\pi y_1) + \sum_{i=1}^{n=1} (y_i - 1)^2[1 + 10\sin^2(\pi y_i + 1)] + (y_n - 1)^2\} + \sum_{i=1}^{30} u(x_i, 10, 100, 4)$	30	$[-50, 50]^D$	0
Penalized2	$f_{13}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{29} (x_i - 1)^2 p[1 + \sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + \sin^2(2\pi x_{30})]\} + \sum_{i=1}^{30} u(x_i, 5, 10, 4)$	30	$[-50, 50]^D$	0

Table A3. Low-dimension Multimodal test functions.

Name	Function	D	Range	f_{opt}
Foxholes	$f_{14}(x) = [\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{j=1}^{2} (x_i - a_{ij})^6}]^{-1}$	2	$[-65.536, 65.536]^D$	0.998
Kowalik	$f_{15}(x) = \sum_{i=1}^{11} \left\lvert a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right\rvert^2$	4	$[-5, 5]^D$	3.075×10^{-4}
Six Hump Camel	$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]^D$	-1.036
Branin	$f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10$	2	$[-5, 10] \times [0, 15]$	0.398
GoldStein-Price	$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 + 1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]^D$	3
Hartman 3	$f_{19}(x) = -\sum_{i=1}^{4} \exp[-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2]$	3	$[0, 1]^D$	-3.86
Hartman 6	$f_{20}(x) = -\sum_{i=1}^{4} \exp[-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2]$	6	$[0, 1]^D$	-3.322
Shekel 5	$f_{21}(x) = -\sum_{i=1}^{5} \left\lvert (x_i - a_i)(x_i - a_i)^T + c_i \right\rvert^{-1}$	4	$[0, 10]^D$	-10.1532
Shekel 7	$f_{22}(x) = -\sum_{i=1}^{7} \left\lvert (x_i - a_i)(x_i - a_i)^T + c_i \right\rvert^{-1}$	4	$[0, 10]^D$	-10.4028
Shekel 10	$f_{23}(x) = -\sum_{i=1}^{10} \left\lvert (x_i - a_i)(x_i - a_i)^T + c_i \right\rvert^{-1}$	4	$[0, 10]^D$	-10.5364

References

1. Lu, R.R.; Wang, J.Y.; Fu, X.T.; Lin, S.H.; Wang, Q.; Zhang, B. Performance analysis and optimization for UAV-based FSO communication systems. *Phys. Commun.* **2022**, *51*, 101594. [CrossRef]

2. Beus, M.; Krpan, M.; Kuzle, I.; Pandžić, H.; Parisio, A. A model predictive control approach to operation optimization of an ultracapacitor bank for frequency control. *IEEE Trans. Energy Convers.* **2021**, *36*, 1743–1755. [CrossRef]
3. Chang, C.; Han, M. Production scheduling optimization of prefabricated building components based on dde algorithm. *Math. Probl. Eng.* **2021**, *2021*, 6672753. [CrossRef]
4. Hu, G.; Du, B.; Wang, X.; Wei, G. An enhanced black widow optimization algorithm for feature selection. *Knowl.-Based Syst.* **2022**, *235*, 107638. [CrossRef]
5. Yoo, S.; Oh, S. Flow analysis and optimization of a vertical axis wind turbine blade with a dimple. *Eng. Appl. Comp. Fluid Mech.* **2021**, *15*, 1666–1681. [CrossRef]
6. Wang, L.; Cao, Q.; Zhang, Z.; Mirjalili, S.; Zhao, W. Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **2022**, *114*, 105082. [CrossRef]
7. Yildiz, A.R.; Mehta, P. Manta ray foraging optimization algorithm and hybrid Taguchi salp swarm-Nelder–Mead algorithm for the structural design of engineering components. *Mater. Test.* **2020**, *64*, 706–713. [CrossRef]
8. Zeinalzadeh, A.; Pakatchian, M.R. Evaluation of novel-objective functions in the design optimization of a transonic rotor by using deep learning. *Eng. Appl. Comp. Fluid Mech.* **2021**, *15*, 561–583. [CrossRef]
9. Pei, H.; Cui, Y.; Kong, B.; Jiang, Y.; Shi, H. Structural parameters optimization of submerged inlet using least squares support vector machines and improved genetic algorithm-particle swarm optimization approach. *Eng. Appl. Comp. Fluid Mech.* **2021**, *15*, 503–511. [CrossRef]
10. Hu, G.; Zhong, J.; Du, B.; Wei, G. An enhanced hybrid arithmetic optimization algorithm for engineering applications. *Comput. Meth. Appl. Mech. Eng.* **2022**, *394*, 114901. [CrossRef]
11. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]
12. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1992.
13. Koza, J.R. Genetic programming as a means for programming computers by natural selection. *Stat. Comput.* **1994**, *4*, 87–112. [CrossRef]
14. Huning, A. Evolutionsstrategie. optimierung technischer systeme nach prinzipien der biologischen evolution. *Arch. Philos. Law Soc. Philos.* **1976**, *62*, 298–300.
15. David, B.F. Artificial Intelligence through Simulated Evolution. In *Evolutionary Computation: The Fossil Record*; Wiley-IEEE Press: Hoboken, NJ, USA, 1998; pp. 227–296. [CrossRef]
16. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
17. Zhao, W.; Wang, L.; Zhang, Z. Atom search optimization and its application to solve a hydrogeologic parameter estimation problem. *Knowl.-Based Syst.* **2019**, *163*, 283–304. [CrossRef]
18. Xing, B.; Gao, W.J. Electromagnetism-like Mechanism Algorithm. In *Innovative Computational Intelligence: A Rough Guide to 134 Clever Algorithms*; Intelligent Systems Reference Library; Springer: Cham, Switzerland, 2014; Volume 62, pp. 347–354. [CrossRef]
19. Formato, R.A. Central force optimization: A new metaheuristic with applications in applied electromagnetics. *Prog. Electromagn. Res.* **2007**, *77*, 425–491. [CrossRef]
20. Pál, K.F. Hysteretic optimization for the Sherrington-Kirkpatrick spin glass. *Phys. A Stat. Mech. Its Appl.* **2006**, *367*, 261–268. [CrossRef]
21. Kaveh, A.; Talatahari, S. A novel heuristic optimization method: Charged system search. *Acta Mech.* **2010**, *213*, 267–289. [CrossRef]
22. Pereira, J.L.J.; Francisco, M.B.; Diniz, C.A.; Oliver, G.A.; Cunha, S.S., Jr.; Gomes, G.F. Lichtenberg algorithm: A novel hybrid physics-based meta-heuristic for global optimization. *Expert Syst. Appl.* **2021**, *170*, 114522. [CrossRef]
23. Hashim, F.A.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W.; Mirjalili, S. Henry gas solubility optimization: A novel physics-based algorithm. *Futur. Gener. Comp. Syst.* **2019**, *101*, 646–667. [CrossRef]
24. Kashan, A.H. A new metaheuristic for optimization: Optics inspired optimization (OIO). *Comput. Oper. Res.* **2015**, *55*, 99–125. [CrossRef]
25. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, IEEE, Nagoya, Japan, 4–6 October 1995; pp. 39–43. [CrossRef]
26. Passino, K.M. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst. Mag.* **2002**, *22*, 52–67. [CrossRef]
27. Zhao, W.; Zhang, Z.; Wang, L. Manta ray foraging optimization: An effective bio-inspired optimizer for engineering applications. *Eng. Appl. Artif. Intell.* **2020**, *87*, 103300. [CrossRef]
28. Li, M.D.; Zhao, H.; Weng, X.W.; Han, T. A novel nature-inspired algorithm for optimization: Virus colony search. *Adv. Eng. Softw.* **2016**, *92*, 65–88. [CrossRef]
29. Yang, X.S.; Deb, S. Cuckoo search via Levy flights. In Proceedings of the World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), Coimbatore, India, 9–11 December 2009; IEEE Publications: Piscataway, NJ, USA; pp. 210–214. [CrossRef]
30. Zhao, W.; Wang, L.; Zhang, Z. Artificial ecosystem-based optimization: A novel nature-inspired meta-heuristic algorithm. *Neural Comput. Appl.* **2020**, *32*, 9383–9425. [CrossRef]
31. Alimoradi, M.; Azgomi, H.; Asghari, A. Trees social relations optimization algorithm: A new Swarm-Based metaheuristic technique to solve continuous and discrete optimization problems. *Math. Comput. Simul.* **2022**, *194*, 629–664. [CrossRef]

32. Xie, L.; Han, T.; Zhou, H.; Zhang, Z.R.; Han, B.; Tang, A. Tuna swarm optimization: A novel swarm-based metaheuristic algorithm for global optimization. *Comput. Intell. Neurosci.* **2021**, *2021*, 9210050. [CrossRef]
33. Peña-Delgado, A.F.; Peraza-Vázquez, H.; Almazán-Covarrubias, J.H.; Torres Cruz, N.; García-Vite, P.M.; Morales-Cepeda, A.B.; Ramirez-Arredondo, J.M. A novel bio-inspired algorithm applied to selective harmonic elimination in a three-phase eleven-level inverter. *Math. Probl. Eng.* **2020**, *2020*, 8856040. [CrossRef]
34. Zhao, W.; Wang, L.; Zhang, Z. Supply-Demand-Based Optimization: A Novel Economics-Inspired Algorithm for Global Optimization. *IEEE Access* **2019**, *7*, 73182–73206. [CrossRef]
35. Ali, A.M.; Nasrat, L.; Hassan, M.H.; Elsayed, S.K.; Kamel, S. Supply Demand-Based Optimization Algorithm for Estimating Break Down Voltage of Silicon Rubber Insulators. In Proceedings of the 2021 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), Valparaíso, Chile, 6–9 December 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–9. [CrossRef]
36. Ginidi, A.R.; Shaheen, A.M.; El-Sehiemy, R.A.; Elattar, E. Supply demand optimization algorithm for parameter extraction of various solar cell models. *Energy Rep.* **2021**, *7*, 5772–5794. [CrossRef]
37. Alturki, F.A.; Al-Shamma'a, A.A.; Farh, H.M.; AlSharabi, K. Optimal sizing of autonomous hybrid energy system using supply-demand-based optimization algorithm. *Int. J. Energy Res.* **2021**, *45*, 605–625. [CrossRef]
38. Jing, C.; Wang, H.; Li, H. Deformation Prediction of Foundation Pit Based on Exponential Power Product Model of Improved Algorithm. *Geofluids* **2021**, *2021*, 7055693. [CrossRef]
39. Ibrahim, S.A.; Kamel, S.; Hassan, M.H.; Elsayed, S.K.; Nasrat, L. Developed Algorithm Based on Supply-Demand-Based Optimizer for Parameters Estimation of Induction Motor. In Proceedings of the 2021 IEEE International Conference on Automation/XXIV Congress of the Chilean Association of Automatic Control (ICA-ACCA), Valparaíso, Chile, 22–26 March 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6.
40. Vanchinathan, K.; Selvaganesan, N. Adaptive fractional order PID controller tuning for brushless DC motor using artificial bee colony algorithm. *Results Control Optim.* **2021**, *4*, 100032. [CrossRef]
41. Karahan, O. Design of optimal fractional order fuzzy PID controller based on cuckoo search algorithm for core power control in molten salt reactors. *Prog. Nucl. Energy* **2021**, *139*, 103868. [CrossRef]
42. Munagala, V.K.; Jatoth, R.K. Improved fractional PIλDμ controller for AVR system using Chaotic Black Widow algorithm. *Comput. Electr. Eng.* **2022**, *97*, 107600. [CrossRef]
43. Kahraman, H.T.; Aras, S.; Gedikli, E. Fitness-distance balance (FDB): A new selection method for meta-heuristic search algorithms. *Knowl.-Based Syst.* **2020**, *190*, 105169. [CrossRef]
44. Kati, M.; Kahraman, H.T. Improving supply-demand-based optimization algorithm with fdb method: A comprehensive research on engineering design problems. *Mühendislik Bilimleri Ve Tasarım Derg.* **2020**, *8*, 156–172. [CrossRef]
45. Hu, G.; Chen, L.; Wang, X.; Wei, G. Differential Evolution-Boosted Sine Cosine Golden Eagle Optimizer with Lévy Flight. *J. Bionic Eng.* **2022**, *2022*, 1–36. [CrossRef]
46. Lai, V.; Huang, Y.F.; Koo, C.H.; Ahmed, A.N.; El-Shafie, A. Optimization of reservoir operation at Klang Gate Dam utilizing a whale optimization algorithm and a Lévy flight and distribution enhancement technique. *Eng. Appl. Comp. Fluid Mech.* **2021**, *15*, 1682–1702. [CrossRef]
47. Liu, J.; Shi, J.; Hao, F.; Dai, M. A novel enhanced global exploration whale optimization algorithm based on Lévy flights and judgment mechanism for global continuous optimization problems. *Eng. Comput.* **2022**, *2022*, 1–29. [CrossRef]
48. Mantegna, R.N. Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes. *Phys. Rev. E* **1994**, *49*, 4677–4683. [CrossRef] [PubMed]
49. Viswanathan, G.M.; Afanasyev, V.; Buldyrev, S.V.; Murphy, E.J.; Prince, P.A.; Stanley, H.E. Lévy flight search patterns of wandering albatrosses. *Nature* **1996**, *381*, 413–415. [CrossRef]
50. Higashi, N.; Iba, H. Particle swarm optimization with Gaussian mutation. In Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706), Indianapolis, IN, USA, 26 April 2003; pp. 72–79. [CrossRef]
51. Gharehchopogh, F.S. An Improved Tunicate Swarm Algorithm with Best-random Mutation Strategy for Global Optimization Problems. *J. Bionic Eng.* **2022**, *19*, 1177–1202. [CrossRef]
52. Alireza, A.L.F.I. PSO with adaptive mutation and inertia weight and its application in parameter estimation of dynamic systems. *Acta Autom. Sin.* **2011**, *37*, 541–549. [CrossRef]
53. Lu, H.; Sriyanyong, P.; Song, Y.H.; Dillon, T. Experimental study of a new hybrid PSO with mutation for economic dispatch with non-smooth cost function. *Int. J. Electr. Power Energy Syst.* **2010**, *32*, 921–935. [CrossRef]
54. Tang, Z.; Tao, S.; Wang, K.; Lu, B.; Todo, Y.; Gao, S. Chaotic Wind Driven Optimization with Fitness Distance Balance Strategy. *Int. J. Comput. Intell. Syst.* **2022**, *15*, 46. [CrossRef]
55. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
56. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [CrossRef]
57. 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]
58. Friedman, M. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *J. Am. Stat. Assoc.* **2012**, *32*, 674–701. [CrossRef]

59. Zhao, W.; Shi, T.; Wang, L.; Cao, Q.; Zhang, H. An adaptive hybrid atom search optimization with particle swarm optimization and its application to optimal no-load PID design of hydro-turbine governor. *J. Comput. Des. Eng.* **2021**, *8*, 1204–1233. [CrossRef]
60. Hemeida, M.G.; Ibrahim, A.A.; Mohamed, A.A.A.; Alkhalaf, S.; El-Dine, A.M.B. Optimal allocation of distributed generators DG based Manta Ray Foraging Optimization algorithm (MRFO). *Ain Shams Eng. J.* **2021**, *12*, 609–619. [CrossRef]
61. Chen, C.; Mo, C. A method for intelligent fault diagnosis of rotating machinery. *Digit. Signal Prog.* **2004**, *14*, 203–217. [CrossRef]
62. Liu, J.; Li, D.; Wu, Y.; Liu, D. Lion swarm optimization algorithm for comparative study with application to optimal dispatch of cascade hydropower stations. *Appl. Soft. Comput.* **2020**, *87*, 105974. [CrossRef]
63. Lu, P.; Zhou, J.; Wang, C.; Qiao, Q.; Mo, L. Short-term hydro generation scheduling of Xiluodu and Xiangjiaba cascade hydropower stations using improved binary-real coded bee colony optimization algorithm. *Energy Conv. Manag.* **2015**, *91*, 19–31. [CrossRef]
64. Liu, J.; Liu, X.; Wu, Y.; Yang, Z.; Xu, J. Dynamic multi-swarm differential learning Harris Hawks Optimizer and its application to optimal dispatch problem of cascade hydropower stations. *Knowl.-Based Syst.* **2022**, *242*, 108281. [CrossRef]