

Article

# Parameter Estimation of Water Quality Models Using an Improved Multi-Objective Particle Swarm Optimization

Yulin Wang <sup>1,2</sup> , Zulin Hua <sup>1,2,\*</sup> and Liang Wang <sup>1,2,3,\*,†</sup>

<sup>1</sup> Key Laboratory of Integrated Regulation and Resource Development on Shallow Lake of Ministry of Education, College of Environment, Hohai University, Nanjing 210098, China; wangyulin01986@126.com

<sup>2</sup> National Engineering Research Center of Water Resources Efficient Utilization and Engineering Safety, Hohai University, Nanjing 210098, China

<sup>3</sup> School of Hydraulic, Energy and Power Engineering, Yangzhou University, Yangzhou 22500, China

\* Correspondence: zulinhua@hhu.edu.cn (Z.H.); right628@126.com (L.W.);  
Tel.: +86-25-8378-7189 (Z.H. & L.W.)

† All the authors contributed equally to this work.

Received: 28 October 2017; Accepted: 29 December 2017; Published: 3 January 2018

**Abstract:** Water quality models are of great importance for developing policies to control water pollution, with the model parameters playing a decisive role in the simulation results. It is necessary to introduce estimation through multi-objective parameters, which is often affected by noise in the data, into water quality models. This paper presents a multi-objective particle swarm optimization algorithm, which is based on the Mahalanobis distance operation, mechanism of cardinality preference and advection-diffusion operator. The Mahalanobis distance operation can effectively reduce the influence of noise in the data on model calibration. The mechanism of cardinality preference and the use of the advection-diffusion operator can prevent non-dominated solutions from falling into the local optimum. Four cases were used to test the proposed approach. The first two cases with true Pareto fronts show that this approach can accurately estimate the true Pareto front with a good distribution, even in the presence of noise. Furthermore, the application of the approach was tested by the O'Connor model and Crops of Engineers Integrated Compartment Water Quality Model. We show that our approach can produce satisfactory results for the multi-objective calibration of complex water quality models. In general, the proposed approach can provide accurate and efficient parameter estimation in water quality models.

**Keywords:** water quality model; improved multi-objective particle swarm optimization; parameter estimation; anti-noise performance

---

## 1. Introduction

Water quality models play a very important role in water quality research that assesses ecosystem health and aids in the development of water pollution control strategies [1,2]. Most water quality models are characterized by differential equations with multiple indicators and a large number of parameters [3,4]. These parameters may be difficult to obtain by experimental measurements or may not even have physical measurements. Furthermore, the selection of these parameters plays a decisive role in the simulation results. Therefore, the calibration of models is crucial in the successful application of water quality models [5,6].

In general, model calibration can be manual or automatic. The manual trial-error process is usually inefficient, time consuming and labor intensive. It strongly depends on the experiential knowledge of the modeler. Implementation of this approach is becoming increasingly difficult due to

increases in the dimensions of parameters and number of objective functions [7–9]. The shortcomings of the trial-error approach can be overcome by automatic calibration, which utilizes the performance of the optimization algorithms and computational efficiency of digital computers. This method has become more and more popular with the increase in computation efficiency [10–12]. The most common method for the automatic calibration of water quality models involves the translation of multiple indicators into a single aggregated scalar with weighted sums [5,13,14]. However, these indicators, which reflect the characteristics of the ecosystem from different aspects, may be inconsistent and conflicting. Therefore, more than one objective of the automatic calibration for water quality models should be considered simultaneously. The essence of automatic calibration is transformed into solving multi-objective optimization problems (MOPs) [15–18].

A set of solutions denoted as the Pareto optimal set can be obtained as the result of trade-off among the indicators [19–22]. A variety of multi-objective evolutionary algorithms (MOEAs) have been developed, which are considered as being effective for solving MOPs [23–29]. Multi-objective particle swarm optimization (MOPSO) is a mainstream algorithm of MOEAs [30,31]. It is a highly competitive algorithm because it is easier to implement and attain convergence compared to other evolutionary algorithms. Furthermore, it is especially suitable for solving MOPs [29,32,33].

There are three aspects that should be considered to improve the quality of solutions when using MOPSO to solve MOPs. First, it is important to properly redefine the global best guide (*gbest*) distribution to obtain a set of non-dominated solutions, which will directly affect the search ability and convergence of the algorithm [34,35]. Some methods for updating *gbest* are based on the topological structure of particles, such as the sigma method [35], minimal particle angle method [36], fitness sharing mechanism [37], niching mechanism [38], crowding distance [39] and its deformation [40], solution distribution entropy [41], dynamic neighborhood strategy [42] and decomposition approach [43,44] among others. The second point involves maintaining non-dominated solutions, which are stored in a so-called external repository. These solutions are identified through the search process. The size of the external repository is usually controlled to balance the tradeoff between a good Pareto optimal set and the computational cost [45]. There are many strategies to attain this goal, such as the  $\epsilon$ -dominance method [46,47], fuzzy clustering technique [38], adaptive grid mechanism [28], the measurements of parallel cell distance [48], preference order scheme [49] and so on. The third aspect involves controlling and maintaining the diversity of particles. The methods include the multi-swarm strategy [45,50], adaptive control of vital parameters [40,51], mutation operator [39], local search scheme [52], population recombination strategy [53], jump improved operator and proportional distribution mechanism [34].

The performance metrics of the MOPSO algorithms are usually based on standard test functions. These algorithms have been widely applied in modeling [10,54,55], although few studies have been examined water quality models [18]. To calibrate water quality models, the observed data are inevitably accompanied by noise. There is a lack of knowledge and validation on whether these methods are able to obtain reasonable results in the case of noise in the data. The noise in the data has a significant impact on the calibration results of water quality models, but the existing MOPSO algorithms do not take these issues into account. Therefore, it is important to note whether the algorithm has an anti-noise performance.

A novel MOPSO algorithm is proposed, which has the purpose of performing parameter estimation of water quality models with additional noise. This algorithm is inspired by mathematical statistics and fluid mechanics theory. The shortcomings of traditional MOPSO algorithms can be overcome using this algorithm. The algorithm is tested on four cases, which include several standard test functions, test functions under different noise strengths and two water quality models.

## 2. Method

The proposed MOPSO algorithm is based on the Mahalanobis distance operation, mechanism of cardinality preference and advection-diffusion operator, so the algorithm was abbreviated as

MCAD-MOPSO. The main aims of the present study are: (1) To reduce the influence of noise in the data on the calibration of water quality models; (2) to maintain the diversity of solutions and prevent the solutions from falling into the local optimum; and (3) to enhance the search ability of algorithms and better guide the search towards the true Pareto front.

The flowchart of the proposed approach is shown in Figure 1. The improved part of the approach is shown in the color block diagram and is further discussed in Sections 2.1–2.3.

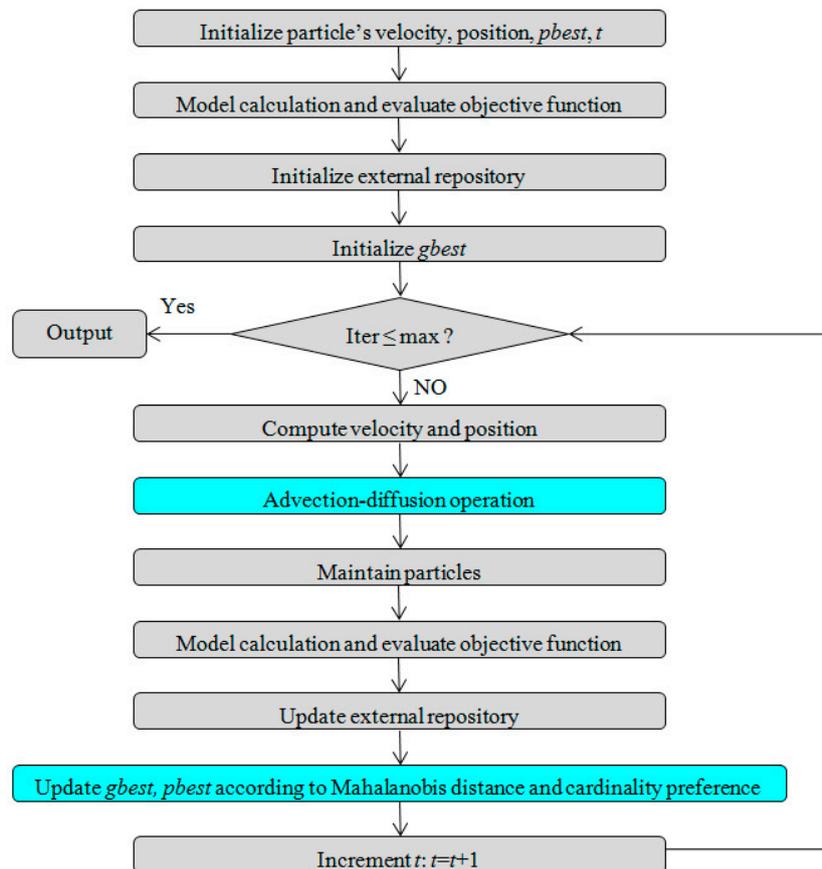


Figure 1. Flowchart of the model calibration with the MCAD-MOPSO.

### 2.1. Mahalanobis Distance Operation

The Mahalanobis distance is widely used in many fields, such as electronics and information systems [56,57], machine learning and pattern recognition [58–61] as well as water quality assessment [62]. The Mahalanobis distance in these applications usually replaces the Euclidean distance for clustering analysis. However, it is less frequently used in the research of MOEAs. The similarity of points in a solution space is usually measured by the Euclidean distance [29,45] and Manhattan distance [39]. For these distance metrics, each objective value of the point is equally important and is considered to be independent from the others. This assumption may not always be satisfied for water quality models. The Mahalanobis distance takes into account unequal variances and correlations among points in the objective space. It has the ability to evaluate distances by assigning different weights or important factors to the points in the objective space [63–65]. Therefore, the Mahalanobis distance is more appropriate for the calibration of water quality models in ecosystems. The distribution information of each function was applied with the covariance matrix of the multi-objective vectors in the Mahalanobis distance, which minimizes the effect of noise in this operation. It is the one of the most important basic features of the MCAD-MOPSO algorithm for the personal best position (*pbest*) selection, *gbest* selection and external repository updating.

**Definition 1** (Mahalanobis distance). The Mahalanobis distance between the two particles in objective space is defined as:

$$d_m(x_i, x_j) = \sqrt{[f(x_i) - f(x_j)]^T S^{-1} [f(x_i) - f(x_j)]} \quad (1)$$

where  $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$  and  $x_j = (x_{j1}, x_{j2}, \dots, x_{jn})$  are independent variables of the optimized function that are marked by the  $i$ th and  $j$ th particles;  $f(x_i) = (f_1(x_i), f_2(x_i), \dots, f_M(x_i))$  and  $f(x_j) = (f_1(x_j), f_2(x_j), \dots, f_M(x_j))$  are objective functions of the  $i$ th and  $j$ th particles;  $T$  means the transposed matrix; and  $S$  is the covariance matrix of the multi-objective functions, which is calculated by the particle's objective functions.

## 2.2. Mechanism of Cardinality Preference

The cardinality preference mechanism was added to update the  $gbest$  and  $pbest$  in the algorithm to ensure that the particles move toward the sparse regions of the search space for maintaining the diversity of particles.

**Definition 2** (Particle inducer). In the particle inducer, we defined  $\forall x_i \in A, y_j \in B, i = 1, 2, \dots, N, j = 1, 2, \dots, L, A \neq \phi, B \neq \phi$ . The inducer of particle  $x_i$  is denoted as  $s_{x_i}$ . For particles in  $B, y_k$  is the most similar particle to  $x_i$ , meaning that  $y_k$  is the inducer of  $x_i$ . Therefore, it is defined as:

$$s_{x_i} = y_k, \text{ while } d_m(x_i, y_k) = \min_{j=1,2,\dots,L} d_m(x_i, y_j), x_i \neq y_j \quad (2)$$

**Definition 3** (Particle induced set). The particle set composed of the particles in set  $A$  induced by  $y_k$  is called the particle  $y_k$  induced set  $I_{y_k}$  on set  $A$ . It is abbreviated as the induced set and is defined as follows:

$$I_{y_k} := \{x_i | s_{x_i} = y_k, i = 1, 2, \dots, N\} \quad (3)$$

The definition of the induced set is based on Mahalanobis distance and reflects the similarity of particles. In a similar way, the set induced by each element in  $B$  can also be obtained. The  $gbest$  and  $pbest$  are updated based on the definition of the particle induced set.

### 2.2.1. Updated $gbest$

The  $gbest$  is updated by the following three steps.

- (1) Compute the cardinality of the induced set of each particle in the external repository  $E$  on the particle population, corresponding to Equations (2) and (3).
- (2) Sort each particle in the external repository according to the cardinality of their induced set in ascending order.
- (3) Randomly select one particle from the specified top portion in the external repository, such as 10%, as the  $gbest$ .

In this way, the non-dominated solutions in the external repository with smaller cardinality are selected as the  $gbest$  of the particles. This strategy enhances the search ability of the algorithm. It also makes it easy for particles to explore the sparse area to prevent the solutions from falling into the local optimum and maintain the diversity of the non-dominated solutions.

### 2.2.2. Updated $Pbset$

The updated  $pbset$  is based on the Pareto dominance relationship between  $pbset$  and the current particle  $x_i$ . Otherwise,  $pbset$  is updated according to the induced set cardinality in the following steps:

- (1) Compare the cardinality of the induced set of the current particle  $x_i$  and its  $pbset$ . The elements in the induced set are on the external repository  $E$ .

- (2) If  $|I_{x_i}| > |I_{pbest}|$ , update  $pbest = x_i$ . Otherwise, do not update  $pbest$ .

The  $pbest$  is updated with a greater particle cardinality, which means that the  $pbest$  becomes closer to the external repository. This strategy can accelerate the convergence rate of the algorithm.

### 2.3. Advection-Diffusion Operator

In the MOPSO algorithm, particles may aggregate after several generations. Therefore, it is difficult for the algorithm to explore potentially better solutions. The advection-diffusion operator was introduced to prevent the solutions from falling into the local optimum.

For each particle with  $n$  dimensions  $x_i = (x_{i1}, x_{i2}, \dots, x_{in})$  in the same induced set  $I_{y_k} (y_k = 1, 2, \dots, |E|)$ , the proposed advection–diffusion of particles is depicted in Figure 2.

- (1) Advection motion

- (a) Arbitrarily select the  $s$ th dimension of each particle in  $I_{y_k}$ .
- (b) The advection magnitude for each  $x_i \in I_{y_k}$  is computed by  $A_k = (u_{kup} - u_{klow}) \cdot C_0 \cdot \left( \frac{|I_{y_k}|}{\sum_{y_k=1}^{|E|} |I_{y_k}|} \right)$ , where  $C_0$  is a constant, while  $u_{kup}$  and  $u_{klow}$  are the upper and lower boundaries of  $x_{is}$ , respectively.

- (2) Diffusion motion

- (a) The diffusion motion  $D_{is}$  for  $s$ th dimension of particle  $x_i$  in  $I_{y_k}$  is determined as  $D_{is} = A_k \cdot random(0,1)$ .
- (b) The magnitude of advection-diffusion for each particle decreases as the number of iterations increases:

$$x_{is} = (D_{is} + A_k) \cdot [1 - (t/maxgen \cdot \beta)]^\alpha + x_{is}, x_i \in I_{y_k} \tag{4}$$

where  $t$  is the number of iterations;  $maxgen$  is the maximum number of iterations; and  $\alpha$  ( $\alpha = 1.5$ ) and  $\beta$  ( $\beta \in (0,1)$ ) are adjustment indices. When  $t \geq maxgen \cdot \beta$ , this operator should not be employed.

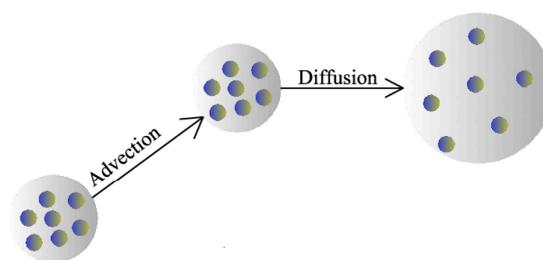


Figure 2. Advection-diffusion of particles in an induced set.

### 2.4. Algorithmic Steps

#### 2.4.1. Initialize

- (a) Initialize the particle’s velocity  $v_i^0 = 0$ , particle’s position  $x_i^0$ , personal best position  $pbest_i^0 = x_i^0$  and iteration counter  $t = 0$
- (b) Evaluate  $f(x_i^0)$
- (c) Initialize the external repository  $E_0$ , storing the non-dominated solutions found in particle population in to  $E$
- (d) Initialize the  $gbest^0$  in accordance with Section 2.2.1.

#### 2.4.2. Repeat Until the Maximum Number of Iterations Is Reached

- (a) Compute the new velocity of each particle using the following formula:

$$v_i^{t+1} = wv_i^t + c_1r_1(pb_{est}_i^t - x_i^t) + c_2r_2(g_{best}^t - x_i^t)$$

where the weight parameter  $w$  is equal to 4. The values of the cognitive parameter  $c_1$  and social parameter  $c_2$  are both 0.5; while  $r_1$  and  $r_2$  are random numbers from 0 to 1. Compute the new position of  $x_i^t$ :

$$x_i^{t+1} = x_i^t + v_i^{t+1}$$

- (b) Use the advection-diffusion operator on each particle, as stated in Section 2.3.
- (c) Maintain the particles within the search space. If decision variable  $x_i$  goes beyond its boundaries, the reflection should be used.
- (d) Evaluate  $f(x_i^{t+1})$
- (e) Update the external repository using the Pareto dominant relationship. If the external repository is full, reduce the non-dominated solutions in the external repository as follows: First, sort the cardinality of the induced set of particles in the external repository in ascending order. Second, a non-dominated solution in the external repository is randomly selected and removed from the specified bottom portion, such as the bottom 10%. It is likely that the non-dominated solutions in dense areas are replaced.
- (f) Update the  $g_{best}^t$  in accordance with Section 2.2.1. Update the  $pb_{est}_i^t$  of each particle in accordance with Section 2.2.2.
- (g) Increment the iteration counter  $t$ .

The computational complexity of the algorithm is dominated by calculating the non-dominated relation of the particles, working out the induced set for particles and sorting the particle cardinality. The objective function computation has  $O(MN)$  complexity if there are  $N$  particles and  $M$  objective functions. Therefore, the complexity of non-dominated relation computation is  $O(MN^2)$  with  $N$  particles in archive. The main complexity of working out the induced set for particles involves the search of the minimum Mahalanobis distance, which requires a computational cost of  $O(N^2)$ . The complexity of sorting the particle cardinality is  $O(N\log 2N)$ . In summary, the overall complexity of the proposed algorithm is  $O(MN^2)$ , which is the same as the MOPSO-CD [39] and NSGA-II algorithms [23] and being slightly more complex than MOPSO [28].

### 3. Results and Discussion

The proposed approach is tested by four cases. Case I is based on several test functions that have a true Pareto front. Several performance metrics from the standard literature are also shown, such as generational distance, error ratio and spacing. Case II adds noise of different strengths in test functions to test our approach's ability to reduce the influence of noise. The applicability of our approach is further tested in Case III, which is the O'Connor model, and Case IV, which is the Crops Engineers Integrated Compartment Water Quality Model (CE-QUAL-ICM) in Chaohu Lake.

#### 3.1. Case I

Multi-objective optimization test problems are crucial in determining the effectiveness of the detection algorithm in solving the multi-objective optimization problems. Currently, some well-known test functions have been widely used in the testing of various multi-objective algorithms, such as ZDT [66], DTLZ [67] and so on. ZDT was proposed by Zitzler et al. and contains six problems (ZDT1–ZDT6). These problems have different characteristics of the Pareto front. ZDT1–ZDT6 are based on two objective functions to reflect essential aspects of multi-objective optimization.

DTLZ1–DTLZ7 were proposed by Deb et al., which can have more than two objective functions to test the ability of algorithms to optimize objective functions.

Several test functions, such as ZDT1–ZDT3, ZDT6, DTLZ1 and DTLZ2, were adopted to test the performance of algorithms in this present paper. ZDT1 has a convex Pareto front, while ZDT2 has a non-convex Pareto front. The Pareto front of ZDT3 is characterized by five discontinuous convex regions. ZDT6 has a complex Pareto front, which is a discontinuous non-convex region with a non-uniform distribution of the solution. Therefore, the test problem for ZDT6 is more difficult than that for ZDT1–ZDT3. DTLZ1 has a linear Pareto front with more than two objective functions. Furthermore, the search space contains multiple local Pareto fronts, which makes it difficult for the multi-objective optimization algorithm to achieve the global optimization. DTLZ2 has a spherical non-linear Pareto front. These test functions have different characteristics of Pareto fronts, which examine the capability of multi-objective algorithms to find and produce a quality distribution of the Pareto front.

There are three performance metrics (generational distance, spacing and error ratio) for multi-objective optimization, which are selected to compare the performance of the proposed algorithm to other popular multi-objective algorithms, such as MOPSO [28], NSGA-II [23] and MOPSO-CD [39]. The parameter setting for each algorithm is shown in Table 1. The parameter dimensions for ZDT1–ZDT3, ZDT6 were set to 30 and the number of objective functions was 2. For the DTLZ1 and DTLZ2, the parameter dimensions and the number of objective functions were 10 and 3, respectively.

**Table 1.** Parameter setting of the multi-objective algorithms.

Parameter	MOPSO	NSGA-II	MOPSO-CD	Proposed Algorithm
Population size	100	100	100	100
Size of external repository	250	250	250	250
Mutation rate	0.5	1/n	0.5	N/A
Cross-over rate	N/A	0.9	N/A	N/A
Cell division	30	N/A	N/A	N/A
Advection-diffusion operator particles	N/A	N/A	N/A	100

The following are the results of 30 independent runs of each algorithm. The true Pareto front and median results with respect to the generational distance metric produced by the proposed algorithm for the six test functions are shown in Figure 3. The comparison of results of the different performance metrics of four algorithms are shown in Table 2.

The generational distance (*GD*) [68] is used to estimate how far the members in the non-dominated set are from their nearest members in the Pareto optimal set. It is measured in objective space and is defined as:

$$GD = \frac{\sqrt{\sum_{i=1}^s d_i^2}}{s} \quad (5)$$

where *s* is the number of members in the non-dominated set found thus far; and *d<sub>i</sub>* is the Euclidean distance between the members in the non-dominated set and their nearest members in the Pareto optimal set. If *GD* is 0, all non-dominated solutions obtained are in the Pareto optimal set. In Table 2, the average performance of the MOPSO-CD and our approach with respect to *GD* is far better than that of NSGA-II and MOPSO for the six test functions. For ZDT1 and ZDT2, the average performance of our approach is equivalent to that of MOPSO-CD with respect to *GD*. For ZDT3 and ZDT6, our approach has the best average performance of the four algorithms, with *GD*s of 0.0011 and 0.0181, respectively. The results of our approach are also best for all test functions with three objects, especially DTLZ1. Furthermore, Table 2 and Figure 3 shows that our approach is able to find good solutions that are accurate estimates of the true Pareto optimal set. The *GD* can be applied to estimate the convergence of the optimal iterations [28,68]. When there is a small difference of *GD* values between the front and back generation, the results can be considered as having converged. The threshold of this difference is set to be 5% times the *GD* of this generation. Overall, the convergence of proposed algorithms is about

the same as MOPSO-CD, while it is slightly better than MOPSO and NSGA-II. This is especially the case in complexity test functions, such as ZDT6, DTLZ1 and DTLZ2.

The error ratio (*ER*) [28] indicates the percentage of solutions (from the non-dominated set obtained so far) that are not members of the Pareto optimal set and is defined as:

$$ER = \frac{\sqrt{\sum_{i=1}^s e_i}}{s} \tag{6}$$

where *s* is the numbers of members in the non-dominated set found thus far. If  $e_i = 1$ , it indicates that the solution *i* is the member of the Pareto optimal set, while  $e_i = 0$  otherwise. A value of 0 for *ER* indicates that all non-dominated solutions obtained by an algorithm belong to the Pareto optimal set. Table 2 shows that all the non-dominated solutions obtained by the proposed approach so far are almost in the Pareto optimal set within a certain precision (<0.01) for most of the test functions. The average performance of NSGA-II and MOPSO with respect to *ER* is equal to 1 for the six test functions. Essentially, these two algorithms may fail to find the true Pareto optimal set for the six test functions. For ZDT1–ZDT3 and DTLZ2 the *ER* evaluation of our approach is 0. In addition, for ZDT6, the *ER* evaluation of our approach is 0.0094, while MOPSO-CD has a value of 0.0124. These results indicate that MOPSO-CD is better than NSGA-II and MOPSO for evaluating *ER*, but is slightly inferior to our proposed approach. This is also seen in the *ER* results of DTLZ1. These results indicate that our approach has better performance than the other three algorithms because they have relatively larger *ER* values than our approach in most test functions.

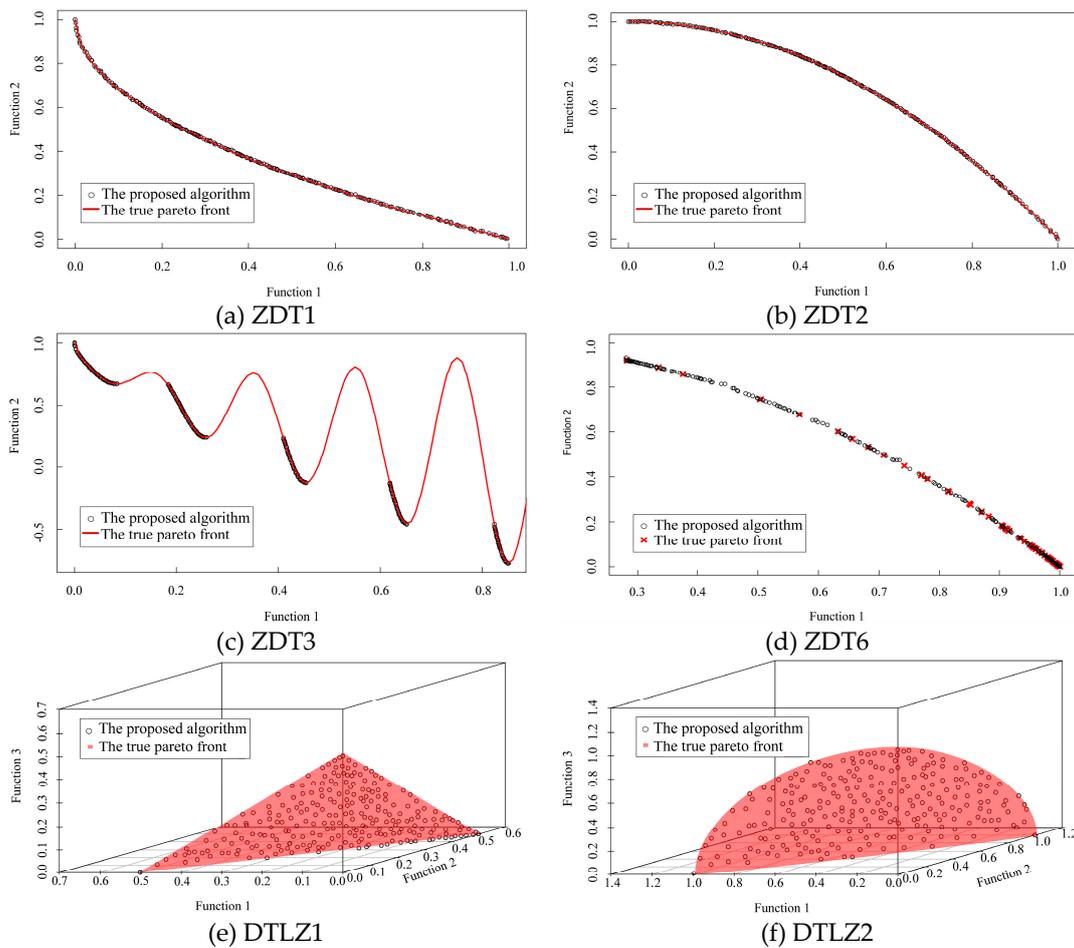


Figure 3. Pareto front produced by the proposed algorithm for the six test functions.

**Table 2.** Algorithm performance metrics for the six test functions.

Test	Metrics		NSGA-II	MOPSO	MOPSO-CD	Proposed Algorithm
ZDT1	GD	Average	0.1052	0.0615	0.0003	0.0005
		Std. Dev.	0.0738	0.0153	<0.0001	0.0006
	ER	Average	1	1	0	0
		Std. Dev.	0	0	0	0
	SP	Average	0.0131	0.0072	0.0051	0.0079
		Std. Dev.	0.0024	0.0013	0.0008	0.0012
ZDT2	GD	Average	0.1273	0.0859	0.0003	0.0003
		Std. Dev.	0.0106	0.0127	<0.0001	<0.0001
	ER	Average	1	1	0	0
		Std. Dev.	0	0	0	0
	SP	Average	0.0074	0.0016	0.0035	0.0083
		Std. Dev.	0.0063	0.0014	0.0002	0.0021
ZDT3	GD	Average	0.1425	0.1104	0.0016	0.0011
		Std. Dev.	0.0036	0.0049	<0.0001	<0.0001
	ER	Average	1	1	0.0002	0
		Std. Dev.	0	0	0.0005	0
	SP	Average	0.0127	0.0131	0.0036	0.0054
		Std. Dev.	0.0026	0.0028	0.0003	0.0028
ZDT6	GD	Average	0.8325	0.5462	0.0200	0.0181
		Std. Dev.	0.0127	0.0243	0.0166	0.0140
	ER	Average	1	1	0.0124	0.0094
		Std. Dev.	0	0	0.0066	0.0051
	SP	Average	0.1532	0.1472	0.0871	0.0985
		Std. Dev.	0.1260	0.0911	0.0881	0.0991
DTLZ1	GD	Average	0.1673	0.1263	0.0771	0.0195
		Std. Dev.	0.6756	0.6332	0.5746	0.0573
	ER	Average	1	1	0.2211	0.2087
		Std. Dev.	0	0	0.3751	0.3957
	SP	Average	0.2805	0.2501	0.0606	0.1291
		Std. Dev.	0.8314	0.8178	0.2305	0.5983
DTLZ2	GD	Average	0.1121	0.1293	0.0763	0.0731
		Std. Dev.	0.0352	0.0455	0.0022	0.0025
	ER	Average	1	1	0	0
		Std. Dev.	0	0	0	0
	SP	Average	0.6356	0.5911	0.3505	0.4203
		Std. Dev.	0.3176	0.1687	0.0335	0.0544

Spacing (*SP*) [69] is used to measure the distribution of solutions throughout the set of non-dominated solutions and is defined as:

$$SP = \sqrt{\frac{1}{s-1} \sum_{i=1}^s (\bar{d} - d_i)^2} \quad (7)$$

where  $d_i = \min_j (\sum_{k=1}^m |f_m^i - f_m^j|)$ ,  $i, j = 1, 2, \dots, s$ ;  $s$  is the number of members in the non-dominated set found so far;  $m$  is the numbers of objectives; and  $\bar{d}$  is the mean of  $d_i$ . If *SP* is 0, it indicates that all of the currently available members of the Pareto front are equidistantly spaced. In Table 2, although the average performance of the proposed algorithm for *SP* is better than those of NSGA-II and MOPSO in some of the test functions, including DTLZ1 and DTLZ2, the results of our approach for the ranking

of  $SP$  are lower than the results of MOPSO-CD. The reason for this difference may be because our approach is based on the Mahalanobis distance for measuring the degree of similarity between the solutions. Therefore, there is a natural inconsistency with  $SP$ , which is based on the Euclidean distance for measuring the equidistance of the solutions. In addition, if the non-dominated solutions produced by the algorithm are not part of the true Pareto front, the  $SP$  metric is irrelevant [28]. For an example, the  $SP$  for the MOPSO is 0.0016, which is better than the  $SP$  value of our algorithm (0.0083) in ZDT2. However, the  $ER$  for the MOPSO is 1, which indicates that the non-dominated solutions obtained by MOPSO are far from the true Pareto front.

Based on the above discussion, it is shown that the proposed approach can generate a better set of non-dominated solutions for the six test functions.

### 3.2. Case II

For testing the ability of the proposed approach in minimizing the influence of noise, different noise strengths from Gaussian distributions  $N(0, 0.05)$  and  $N(0, 0.15)$  were added to the test functions in Case II. The setting of the algorithm-related parameters is the same as in Case I. The comparison of the algorithm results is shown in Tables 3 and 4.

Table 3 shows that the average performance of the two algorithms in all of the test functions with noise  $N(0, 0.05)$  is inferior to that without noise, which is shown in Table 2. However, the proposed algorithm has an overwhelming advantage compared to MOPSO-CD in all of the performance tests. The average performance of our approach with respect to  $GD$  and  $ER$  for ZDT1 is 0.0079 and 0.0127, respectively, which is 5.4–5.8 times better than that of MOPSO-CD. For ZDT2, the average performance of MOPSO-CD with respect to  $GD$  and  $ER$  is 0.0571 and 0.1302, which is approximately 158% and 350% more than the  $GD$  (0.0221) and  $ER$  (0.0290) of our approach, respectively. The  $GD$  and  $ER$  of the MOPSO-CD for ZDT3 are approximately 6 and 7 times that of our approach, respectively. Compared to other ZDT series test functions, the superior performance of our approach is not as significant in ZDT6, but its corresponding  $GD$ ,  $SP$  and  $ER$  are also approximately 1/3–7/8 of those of MOPSO-CD. In addition, the average performance of our approach with respect to  $SP$  for all of the test functions are better than those of MOPSO-CD. The  $SP$  is 0.0521 in our approach and 0.1571 in MOPSO-CD for ZDT3, which is 70% higher than ours. For DTLZ1 and DTLZ2 test functions, the performance of our algorithm is far better than MOPSO-CD with the results of the MOPSO-CD approach being 2–5 times our results.

Table 3 shows that our approach has better stability compared to MOPSO-CD in all test functions. The standard deviation of the  $GD$  values from the MOPSO-CD for ZDT1 and ZDT3 is approximately 6 times that of our approach, while this difference is around 2–9 times for DTLZ1 and DTLZ2. The above results show that when noise  $N(0, 0.05)$  is added to the test functions, our approach has better results than MOPSO-CD in terms of the average performance metrics and standard deviation, which means that proposed algorithm can effectively reduce the influence of noise on the optimization results.

To test the performance of the algorithm under stronger noise, noise according to  $N(0, 0.15)$  was added to the test functions. The test results of the performance of the two algorithms can be seen in Table 4. The advantages of our approach are not as obvious as in Table 3, but are 1.5–6 times better than the results of MOPSO-CD. This may be because the signal-to-noise ratio decreases with the increase in the disturbance intensity, causing the value to become submerged by noise. Nevertheless, our approach is still superior to MOPSO-CD in almost all of the test functions in Table 4. For instance, the  $ER$  of MOPSO-CD for DTLZ2 is 1, while the  $ER$  of our algorithm is only about 0.5. This indicates that the MOPSO-CD algorithm is hardly to obtain the true Pareto optimal set, while our algorithm was able to obtain this set even under the stronger noise.

The convergence of the proposed algorithm under noise is better than MOPSO-CD. There are 1500–2000 fewer function evaluations from our algorithm compared to MOPSO-CD for ZDT1, ZDT2, ZDT3 and ZDT6 under  $N(0, 0.05)$  noise. For DTLZ1 and DTLZ2, this difference is 1200. The lead of the proposed algorithm is slightly larger under the stronger noise.

Our approach has a good performance compared to MOPSO-CD with respect to *GD*, *SP* and *ER* almost in all of the test functions with different noise strengths. This result indicates that our approach can produce a well-distributed non-dominated solution set even under different strengths of noise. The stability of our approach, which has a smaller standard deviation with respect to *GD*, *SP* and *ER*, is also better than that of MOPSO-CD. Under different noise strengths, our approach has a good performance for convergence, distribution and anti-noise.

**Table 3.** Algorithm performance metrics for the six test functions with noise  $N(0, 0.05)$ .

Test	Metrics		MOPSO-CD	Proposed Algorithm
ZDT1	<i>GD</i>	Average	0.0463	0.0079
		Std. Dev.	0.0571	0.0098
	<i>ER</i>	Average	0.0693	0.0127
Std. Dev.		0.0705	0.0150	
	<i>SP</i>	Average	0.1070	0.0484
		Std. Dev.	0.1094	0.0599
ZDT2	<i>GD</i>	Average	0.0571	0.0221
		Std. Dev.	0.0806	0.0357
	<i>ER</i>	Average	0.1302	0.0290
Std. Dev.		0.1890	0.0331	
	<i>SP</i>	Average	0.1120	0.0652
		Std. Dev.	0.1196	0.0788
ZDT3	<i>GD</i>	Average	0.0864	0.0115
		Std. Dev.	0.0892	0.0153
	<i>ER</i>	Average	0.0985	0.0152
Std. Dev.		0.1052	0.0226	
	<i>SP</i>	Average	0.1571	0.0521
		Std. Dev.	0.2331	0.0716
ZDT6	<i>GD</i>	Average	0.1670	0.0702
		Std. Dev.	0.2418	0.0474
	<i>ER</i>	Average	0.0999	0.0716
Std. Dev.		0.1364	0.0430	
	<i>SP</i>	Average	0.3485	0.2985
		Std. Dev.	0.2438	0.2190
DTLZ1	<i>GD</i>	Average	1.4542	0.5897
		Std. Dev.	4.3173	0.7555
	<i>ER</i>	Average	0.3263	0.0627
Std. Dev.		0.3699	0.0428	
	<i>SP</i>	Average	3.2972	1.0251
		Std. Dev.	9.9006	4.9083
DTLZ2	<i>GD</i>	Average	0.2443	0.1854
		Std. Dev.	0.0223	0.0129
	<i>ER</i>	Average	0.1413	0.0489
Std. Dev.		0.0762	0.0023	
	<i>SP</i>	Average	1.0686	0.5932
		Std. Dev.	0.3935	0.0782

**Table 4.** Algorithm performance metrics for the six test functions with noise  $N(0, 0.15)$ .

Test	Metrics		MOPSO-CD	Proposed Algorithm
ZDT1	GD	Average	0.1616	0.0371
		Std. Dev.	0.1309	0.0294
	ER	Average	0.3446	0.2587
		Std. Dev.	0.1069	0.0945
ZDT2	GD	Average	0.1639	0.0533
		Std. Dev.	0.1528	0.0597
	ER	Average	0.3788	0.2701
		Std. Dev.	0.1961	0.1490
ZDT3	GD	Average	0.1234	0.0419
		Std. Dev.	0.1134	0.0656
	ER	Average	0.3336	0.2465
		Std. Dev.	0.1178	0.0998
ZDT6	GD	Average	0.1797	0.1399
		Std. Dev.	0.2479	0.1692
	ER	Average	0.3846	0.1803
		Std. Dev.	0.2948	0.0984
DTLZ1	GD	Average	0.3838	0.2858
		Std. Dev.	0.1550	0.0882
	ER	Average	0.4545	0.3082
		Std. Dev.	0.2788	0.2264
DTLZ2	GD	Average	6.9200	0.9517
		Std. Dev.	7.1328	1.8960
	ER	Average	0.6430	0.5162
		Std. Dev.	0.5618	0.3785
DTLZ1	GD	Average	12.0001	2.8862
		Std. Dev.	14.8153	5.7169
	ER	Average	0.3847	0.2747
		Std. Dev.	0.0466	0.0247
DTLZ2	GD	Average	1	0.5323
		Std. Dev.	0	0.1236
	ER	Average	2.1736	0.6745
		Std. Dev.	0.5832	0.0955

### 3.3. Case III

The O'Connor model is a one-dimensional steady-state river model based on the Streeter-Phelps model, which takes into account the degradation, sedimentation, oxygen consumption of organic matter decomposition oxygen demand (CBOD) and nitrification oxygen demand (NBOD), as well as the role of dissolved oxygen (DO) re-oxygenation [70]. It contains four parameters: the CBOD degradation coefficient  $K_1$ , re-oxygenation coefficient  $K_2$ , CBOD settlement coefficient  $K_3$  and NBOD degradation coefficient  $K_N$ . The four parameters are a good measure of the performance of the algorithm in model applications, in which these values vary widely. These values have a range of 0.9–4. According to the measured results of a river section on the Taihu Plain in China, the value of the initial conditions and parameters are shown in Table 5.

**Table 5.** The values of the parameters and initial conditions of a river section.

Parameter (/d)				Initial Condition (mg/L)		
$K_1$	$K_2$	$K_3$	$K_N$	CBOD	NBOD	Oxygen Deficit (OD)
0.950	3.530	0.100	0.125	0.305	0.100	0.000

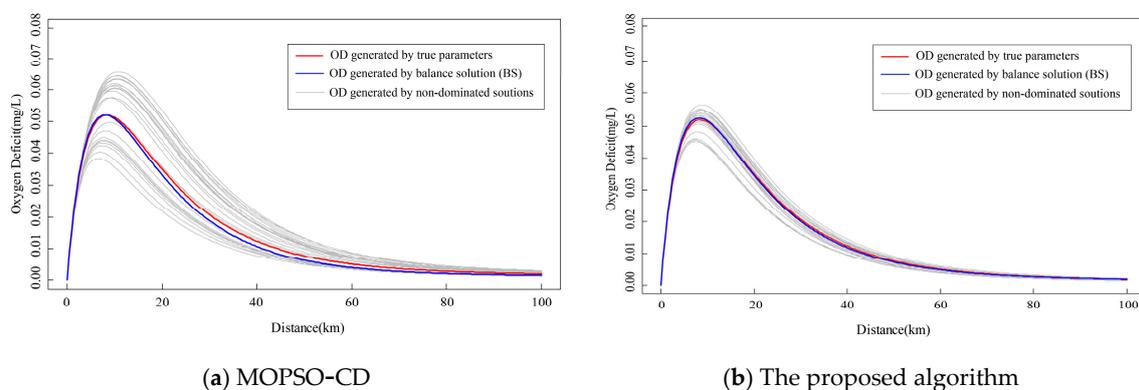
In the following section, the two algorithms are used for parameter estimation of the O'Connor model. The three objection functions, which are the relative errors of CBOD, NBOD and OD, are optimized by MOPSO-CD and the proposed algorithm. The population size, number of iterations and size of the external repository are set to 40, 50 and 100, respectively. The calibration initially occurs in regard to the variables CBOD, NBOD, and OD without noise, with the results shown in Table 6.

**Table 6.** Calibration results of the O'Connor model without noise.

Parameter	MOPSO-CD			Proposed Algorithm		
	Range	Mean	Std. Deviation	Range	Mean	Std. Deviation
$K_1$	0.959–1.060	1.009	0.028	0.946–1.043	0.978	0.025
$K_2$	3.303–4.006	3.652	0.168	3.304–4.399	3.570	0.153
$K_3$	0.070–0.096	0.081	0.008	0.090–0.118	0.105	0.011
$K_N$	0.063–0.080	0.072	0.006	0.071–0.147	0.127	0.023

In Table 6, although the range and standard deviation of the parameters obtained by MOPSO-CD are small, which shows more stability than ours, the  $K_1$ ,  $K_3$  and  $K_N$  obtained do not cover the true parameter values. This means that the parameters from MOPSO-CD are unreliable and cannot be used in modeling. Compared with the MOPSO-CD results, the parameters generated by the proposed algorithm cover these true values and the means of the parameters obtained by our algorithm are close to the true values. For example, the mean of  $K_N$  obtained by our algorithm is 0.127, while it is 0.072 for MOPSO-CD. The relative error for the true values (0.125) are 1.6% and 42.4%, respectively. Since the means of the parameters are often used for calculating the model in practice, the above results indicate the advantage of our algorithm.

Taking OD as an example, the changes in OD (0–100 km) are shown in Figure 4, which correspond to different non-dominated solutions obtained. The line generated by the balance solution (BS) is also shown, which has the smallest sum of the distance to the true parameters.



**Figure 4.** OD prediction from 0 to 100 km. The red line is generated by true parameters. The gray lines are generated by non-dominated solutions, while the blue line is generated by the BS.

Since the relationship between the variables and parameters is non-linear and they have complex correlations with each other, the range of OD values obtained by our approach is narrower than that of

MOPSO-CD even if the parameter range of our approach is larger than that of MOPSO-CD. This result indicates that there is less uncertainty in the results worked by our approach. Although the result of the true parameters is covered by the results of two algorithms, the OD curve generated by the BS of our approach is closer to the true value curve. The maximum error of the curve from the BS of our approach is 1.2%, while the maximum error is approximately 15.6% in MOPSO-CD. The maximum OD position calculated by MOPSO-CD is different from the true position. These results showed that there would be a large ecological risk if the MOPSO-CD results were applied to water quality predictions. Our results are almost identical with the true position at the same time.

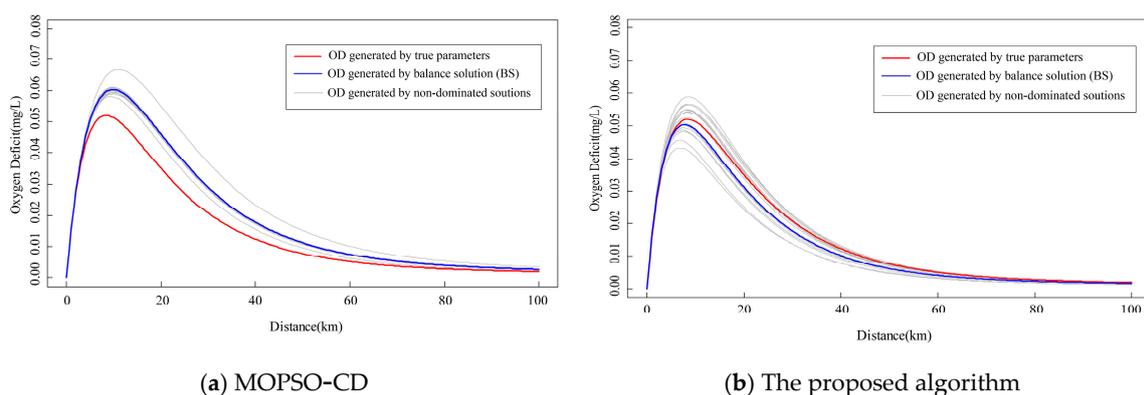
The noise according to the  $N(0, 0.05)$  and  $N(0, 0.15)$  are added to the observed values and the optimum objection functions are the same as the condition without noise. The estimation of the parameters is conducted according to the relative errors of CBOD, NBOD and OD. The calibration results of the two algorithms are shown in Tables 7 and 8. The OD curves with different noise levels are shown in Figures 5 and 6.

**Table 7.** O'Connor model results with noise  $N(0, 0.05)$  multiplied by the calculated values.

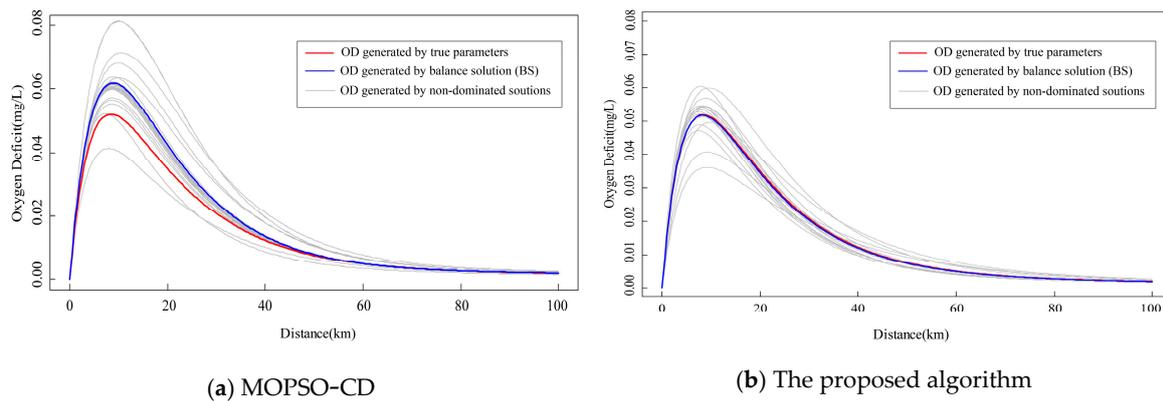
Parameter	MOPSO-CD			Proposed Algorithm		
	Range	Mean	Std. Deviation	Range	Mean	Std. Deviation
$K_1$	0.889–0.941	0.919	0.013	0.924–1.073	1.000	0.043
$K_2$	2.378–2.930	2.779	0.088	3.526–4.351	3.846	0.185
$K_3$	0.084–0.096	0.093	0.003	0.094–0.137	0.118	0.013
$K_N$	0.096–0.164	0.133	0.015	0.089–0.135	0.118	0.011

**Table 8.** O'Connor model results with noise  $N(0, 0.15)$  being used to multiply the calculated values.

Parameter	MOPSO-CD			Proposed Algorithm		
	Range	Mean	Std. Deviation	Range	Mean	Std. Deviation
$K_1$	0.819–1.211	1.053	0.086	0.605–1.170	0.962	0.139
$K_2$	2.138–4.081	2.935	0.517	2.846–4.269	3.609	0.345
$K_3$	0.108–0.143	0.126	0.007	0.074–0.134	0.091	0.013
$K_N$	0.080–0.130	0.096	0.011	0.094–0.182	0.126	0.025



**Figure 5.** OD prediction for 0–100 km under the condition of noise according to  $N(0, 0.05)$  multiplied by the calculated values obtained. The red line is generated by true parameters. The gray lines are generated by non-dominated solutions, while the blue line is generated by BS.



**Figure 6.** OD prediction for 0–100 km under the condition of noise according to  $N(0, 0.15)$  being multiplied by the calculated values obtained. The red line is generated by true parameters. The gray lines are generated by non-dominated solutions, while the blue line is generated by BS.

The range of  $K_1$ ,  $K_2$  and  $K_3$  obtained by MOPSO-CD still does not cover the true value, while the range of parameters obtained by our method covered the true values. In the case of adding noise according to  $N(0, 0.05)$  being used to multiply the calculated values, there are only 8 non-dominated solutions generated by MOPSO-CD. The range of OD values obtained by MOPSO-CD does not cover the true value curve and has large errors (Figure 5). The position and value of the maximum OD for the BS curve are very different from the true value curve with an error of 23%. The shape of the curve determined by BS also diverges from the true value curve. These parameters can hardly be used for model prediction due to the large risk. By contrast, our approach generates more non-dominated solutions and the OD curves completely cover the true value. The BS curve is not as good as that without noise, but the errors of the value and phase are much smaller than those of the MOPSO-CD. Therefore, our approach can be applied in a practical environment with noise.

Due to the increase in noise strength, the range of parameters obtained by MOPSO-CD and our approach expanded, while the range of  $K_3$  by MOPSO-CD failed to cover the true value. The range of the maximum OD calculated by MOPSO-CD is 0.038–0.082, which is significantly larger than that from our approach (0.035–0.059; Figure 6). This result indicates that the uncertainty of the maximum OD generated by MOPSO-CD is higher. The OD curves obtained by the two algorithms cover the true value curve. However, the true value curve is at the lower edge of the OD curve area of MOPSO-CD (Figure 6a). The shape of the OD curves and position of the maximum OD of the MOPSO-CD results have large discrepancies from the true values. By contrast, the true value curve is located in the central region of the OD curves produced by our approach (Figure 6b). For both the shape and position of the maximum OD, the OD curves of our approach are very close to the true value curve. In addition, the curve of BS obtained by MOPSO-CD is very different from the true value curve at the position of the maximum OD. This shows that the results from our approach are better. This also shows that our approach has more advantages in the case of noise.

In general, the O'Connor model calibration results from our approach are consistent with the true parameter values in a water ecosystem with noise, even if the range of parameters is quite different. Its performance and calibration results are superior to those of the classic MOPSO-CD.

### 3.4. Case IV

CE-QUAL-ICM is used in the fourth case. It was developed by the U.S. Army Corps of Engineers Engineer Research Development Center [71]. The model computes physical properties, multiple forms of algae, nitrogen, phosphorus, carbon, silica, chemical oxygen demand, salinity, temperature, metal and dissolved oxygen cycles. It involves interactions between various algae and various nutrients, which takes into account the process of algal growth, metabolism, predation and settling. At present,

CE-QUAL-ICM has been successfully applied to the study of eutrophication processes of various water bodies [72,73]. However, it is considered to be difficult to calibrate due to inevitable data noise and the high dimensionality of the parameter space.

A lake is an open system for material and energy exchange with the environment. We used Chaohu Lake in eastern China as an example to study the parameter optimization problem of the CE-QUAL-ICM model. Chaohu Lake is approximately 780 km<sup>2</sup>, with a depth of 1–7 m. As it has been seriously polluted, it is one of the most eutrophic lakes in China and has suffered from frequent outbreaks of algae in recent decades. Therefore, the eutrophication of Chaohu Lake is a significant concern [74–76]. Nine major rivers around Chaohu Lake were considered in the model calibration. Although the lake is large, a proper water quality model can capture the main characteristics of the water quality changes over time in Chaohu Lake, even when ignoring spatial heterogeneity and hydrodynamic influences [77,78].

The cyanobacteria and DO are two important water quality indicators and were selected for use in the model parameter estimation in the Chaohu Lake water quality model. All data for the modeling were collected from 25 sites that were monitored about weekly in July–September in 2009. It is necessary to select the parameters for the model calibration. The 15 parameters with the greatest influence on the model variables were identified, with their names and ranges provided in Table 9. The other parameters used the default values [71].

**Table 9.** Parameters in the CE-QUAL-ICM model.

Parameter	Value	Unit	Range 1	Range 2	Description
NitM	0.008	G Nm <sup>-3</sup> day <sup>-1</sup>	0.004–0.012	0.007–0.011	maximum nitrification rate
KRP	0.005	day <sup>-1</sup>	0.0025–0.0075	0.0032–0.0069	minimum hydrolysis rate of RPOP
KDN	0.015	day <sup>-1</sup>	0.007–0.023	0.008–0.021	minimum mineralization rate of DON
KHP	0.006	Gp m <sup>-3</sup>	0.003–0.009	0.004–0.007	half-saturation constant for phosphorus uptake for cyanobacteria
BMR	0.15	day <sup>-1</sup>	0.08–0.23	0.08–0.017	basal metabolism rate at reference temperature for cyanobacteria
KTB	0.060	°C <sup>-1</sup>	0.030–0.090	0.031–0.086	effect of temperature on metabolism for cyanobacteria
KTHDR	0.069	°C <sup>-1</sup>	0.035–0.102	0.036–0.095	effect of temperature on hydrolysis of particulate organic matter
KTCOD	0.052	°C <sup>-1</sup>	0.026–0.077	0.029–0.071	effect of temperature on oxidation of COD
KR	0.36	day <sup>-1</sup>	0.18–0.54	0.20–0.38	reaeration coefficient
PRR	0.038	day <sup>-1</sup>	0.019–0.057	0.019–0.035	reference predation rate for cyanobacteria
KLN	0.052	day <sup>-1</sup>	0.026–0.078	0.029–0.068	minimum hydrolysis rate of LPON
KRN	0.005	day <sup>-1</sup>	0.0025–0.0075	0.0027–0.0074	minimum hydrolysis rate of RPON
KDP	0.18	day <sup>-1</sup>	0.09–0.27	0.15–0.22	minimum mineralization rate of DOP
KLP	0.10	day <sup>-1</sup>	0.05–0.15	0.07–0.14	minimum hydrolysis rate of LPOP
KRPALG	0.20	m <sup>3</sup> (gC) <sup>-1</sup> day <sup>-1</sup>	0.10–0.30	0.14–0.28	constant that relates hydrolysis of RPOP to algal biomass

Footnote: Range 1 refers to the calculation range of parameter. Range 2 refers to the range of non-dominated solutions that have been obtained.

The CE-QUAL-ICM model in Chaohu Lake is calibrated with a daily time resolution to estimate the cyanobacteria and DO over three months. The two objection functions,  $OF_1$  and  $OF_2$ , which

are the root mean square error (RMSE) of DO and cyanobacteria, respectively, are optimized by the proposed algorithm.

$$\text{RMSE} = OF_i = \sqrt{\sum_{d=1}^D (O_d - P_d)^2 / D}, \quad i = 1, 2 \quad (8)$$

where  $O$  are the observations;  $P$  are the predictions; and  $D$  is the total number of observations. The ranges of the parameters for the Pareto set is shown in Table 9. The population size, number of iterations and size of the external repository are set to 20, 30 and 50 here, respectively. Table 9 shows that the range 2 obtained for most of the parameters by calibration is significantly less than the original range 1. Hence, the uncertainty in most of the parameters can be reduced by our approach.

The Pareto front in the objective space is shown in Figure 7. There are 15 non-dominated solutions that are obtained by our approach. From the definition of the non-dominated solution set, none of the corresponding tradeoffs can be considered to be better than the others if there is a lack of information by which to judge. However, a decision-maker can choose an appropriate solution based on the importance of the objectives for their own interests. The leftmost point of the Pareto front corresponding to the parameter vector assigns the highest fit to the cyanobacteria calibration at the expense of the weakest DO calibration fit. The rightmost point indicates that the highest importance is assigned to DO. The solution with the solid circle in Figure 7 is a better tradeoff of the calibration of the cyanobacteria and DO. The objection functions  $OF_1$  and  $OF_2$  for the selected solution are equal to 0.80 mg/L and 1.79  $\mu\text{g/L}$ , respectively.

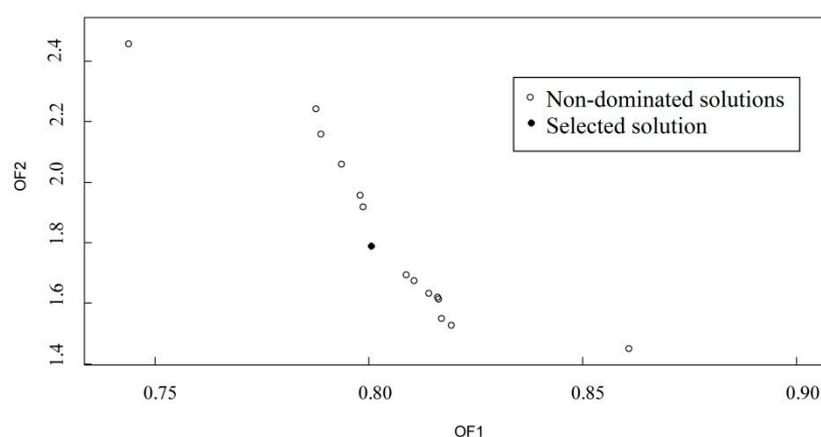
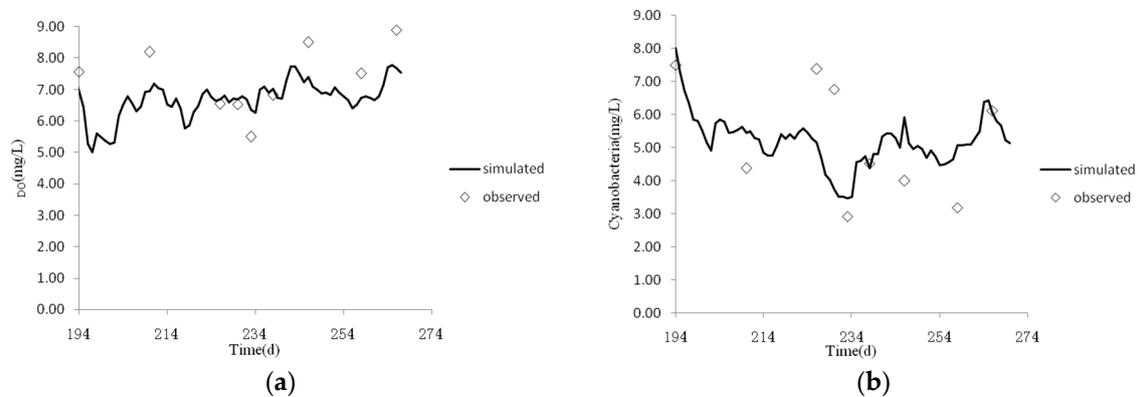


Figure 7. The Pareto front in the objective space.

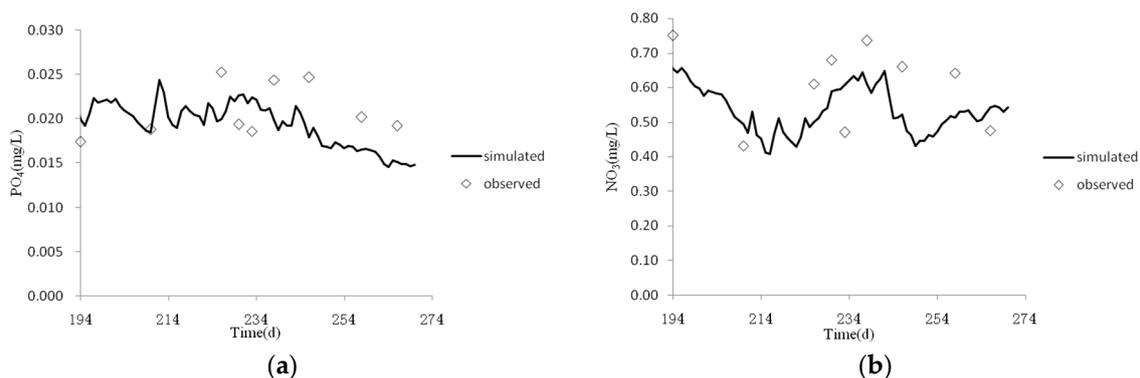
The selected solution is used to simulate the DO and cyanobacteria in summer in Chaohu Lake (Figure 8). Comparing the simulated results of the model with the observed values, the average relative errors of cyanobacteria and DO were approximately 29% and 28%. The Pearson correlation coefficients between the simulated and observed values of cyanobacteria and DO are 0.59 and 0.64, respectively. The simulation results are consistent with the observed values and trends in Chaohu Lake. These correlation coefficients were not very large, suggesting the trade-off between RMSE of DO and cyanobacteria and the possibility of a non-linear relationship between simulated and observed values.

Figure 9 presents the simulation results of phosphate ( $\text{PO}_4$ ) and nitrate ( $\text{NO}_3$ ), which are the parameters of the selected solution. Their average relative errors and Pearson correlation coefficients for  $\text{PO}_4$  and  $\text{NO}_3$  were approximately 24% and 0.62 as well as 27% and 0.66, respectively. The simulation trends of  $\text{PO}_4$  and  $\text{NO}_3$  are also consistent with the observed values. This result shows that the parameters obtained by the calibration of the cyanobacteria and DO can also be used to simulate other environmental variables, such as  $\text{PO}_4$  and  $\text{NO}_3$ . This result indicates that there are correlations among variables in the water quality model.

The above results show that our approach has the ability to obtain good results in the multi-objective calibration of a complex water quality model, even though there is unavoidable observation noise in the ecosystem model.



**Figure 8.** The simulation of cyanobacteria and DO in the summer in Chaohu Lake. (a) DO, (b) Cyanobacteria. The solid line shows the result of the simulation. The diamond box represents the observed value.



**Figure 9.** The simulation of phosphate and nitrates in the summer in Chaohu Lake. (a) PO<sub>4</sub>, (b) NO<sub>3</sub>. The solid line shows the result of the simulation. The diamond box represents the observed value.

#### 4. Conclusions

The water quality models involve problems with multiple objectives and parameters, which increase the difficulty of model calibration. Furthermore, models are often affected by noise in the data. This paper presents a novel algorithm (MCAD-MOPSO) to reduce the impact of noise on the parameter estimation of water quality models. The proposed approach is based on the Mahalanobis distance operation, mechanism of cardinality preference and advection-diffusion operator. The Mahalanobis distance operation can effectively reduce the influence of data noise on the calibration of a water quality model. The mechanism of cardinality preference can use more populated information to enhance the search ability of the algorithm and force the particles to move toward the sparse regions of the search space. The advection-diffusion operator can prevent the solutions from falling into the local optimum, explore potentially better solutions and maintain the diversity of solutions in the external repository.

The first two cases are based on test functions and metrics from the literature, with addition of noise to the test functions. In the absence of noise, the average performance of our approach is significantly better than that of NSGA-II and MOPSO, while it is even better than MOPSO-CD regarding most test functions. This result shows that our approach can produce a good approximation with a good distribution of the true Pareto front in all test functions. When adding the different noise

strengths from the Gaussian distribution to the test functions, the results show that the ability of our approach to reduce noise is better than that of MOPSO-CD. It has a good performance with respect to *GD*, *SP* and *ER* in all of the test functions. The average performance with respect to *GD*, *ER* and *SP* of MOPSO-CD for all of the test functions is approximately 1.3–7.5, 1.4–6.4 and 1.2–3.0 times that of our approach corresponding to the noise of  $N(0,0.05)$ , while this is approximately 1.4–7.0, 1.2–1.8 and 1.2–4.3 times for the noise of  $N(0,0.15)$ , respectively. Even if the effect of noise is larger, our approach has the ability to accurately approximate the true Pareto front with a good distribution.

The application of the proposed algorithm is further tested on two frequently used models, the O'Connor model and CE-QUAL-ICM model, in the ecosystem. The four parameters of the O'Connor model were calibrated by MOPSO-CD and our approach. The results show that regardless of different noise strengths, the range of parameters obtained by our approach covered the true parameter values. The means of the parameters calculated by the proposed approach were closer to their true values. The OD prediction of our approach had less uncertainty, while the BS curve was closer to the true value curve. Moreover, the error of the maximum OD value and position obtained by our approach was much smaller than that of the MOPSO-CD. The performance and calibration results of our approach were superior to those of the classic MOPSO-CD. Finally, in a shallow lake, a complex water quality model (CE-QUAL-ICM) was calibrated using our approach. The results show that our approach performed well in the calibration of the lake ecosystem model with added noise from natural observations. The selected non-dominated solution obtained good results for the simulation of the cyanobacteria biomass, DO as well as the  $PO_4$  and  $NO_3$  concentrations.

In summary, our approach can effectively reduce the influence of noise in data on the multi-objective calibration of the models, which the traditional algorithm does not take into account. We conclude that our approach is a highly competitive algorithm for optimization problems, regardless of the standard test function or the complex model in the ecosystem. Our approach provides a useful exploration for parameter estimation of water quality models.

**Acknowledgments:** This work was financially supported by the National Natural Science Foundation of China (Grant Nos. 51379060, 51739002), the Major Science and Technology Program for Water Pollution Control and Treatment (Grant No. 2012ZX07103-005), the Qing Lan Project and PAPD Project, and the Fundamental Research Funds for the Central Universities (2016B21214, 2015B24714, 2015B36314, 2016B44014, 2015B42014).

**Author Contributions:** Yulin Wang, Zulin Hua and Liang Wang conceived and designed the paper; the paper was written by Yulin Wang; the algorithm is implemented by Liang Wang; Zulin Hua reviewed and improved the manuscript; the data compilation and statistical analysis were completed by all authors.

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

## References

1. Chanudet, V.; Smits, J.; Van Beek, J.; Boderie, P.; Guérin, F.; Serça, D.; Deshmukh, C.; Descloux, S. Hydrodynamic and water quality 3D modelling of the Nam Theun 2 Reservoir (Lao PDR): Predictions and results of scenarios related to reservoir management, hydrometeorology and nutrient input. *Hydroécol. Appl.* **2016**, *19*, 87–118. [[CrossRef](#)]
2. Di Maggio, J.; Fernández, C.; Parodi, E.R.; Diaz, M.S.; Estrada, V. Modeling phytoplankton community in reservoirs. A comparison between taxonomic and functional groups-based models. *J. Environ. Manag.* **2016**, *165*, 31–52. [[CrossRef](#)] [[PubMed](#)]
3. Arhonditsis, G.B.; Brett, M.T. Eutrophication model for Lake Washington (USA): Part I. Model description and sensitivity analysis. *Ecol. Model.* **2005**, *187*, 140–178. [[CrossRef](#)]
4. Yang, L.K.; Peng, S.; Zhao, X.H.; Li, X. Development of a two-dimensional eutrophication model in an urban lake (China) and the application of uncertainty analysis. *Ecol. Model.* **2017**, *345*, 63–74. [[CrossRef](#)]
5. Chaudhary, S.; Dhanya, C.T.; Kumar, A. Sequential calibration of a water quality model using reach-specific parameter Estimates. *Hydrol. Res.* **2017**, *48*, nh2017246. [[CrossRef](#)]
6. Liu, S.; Butler, D.; Brazier, R.; Heathwaite, L.; Khu, S.T. Using genetic algorithms to calibrate a water quality model. *Sci. Total Environ.* **2007**, *374*, 260–272. [[CrossRef](#)] [[PubMed](#)]

7. Hogue, T.S.; Sorooshian, S.; Gupta, H.; Holz, A.; Braatz, D. A multistep automatic calibration scheme for river forecasting models. *J. Hydrometeorol.* **2000**, *1*, 524–542. [[CrossRef](#)]
8. Brett, M.; Arhonditsis, G. Eutrophication model for Lake Washington (USA) part II—Model calibration and system dynamics analysis. *Ecol. Model.* **2005**, *187*, 179–200. [[CrossRef](#)]
9. Mao, J.; Chen, Q.; Chen, Y. Three-dimensional eutrophication model and application to Taihu Lake, China. *J. Environ. Sci.* **2008**, *20*, 278–284. [[CrossRef](#)]
10. Confesor, R.B.; Whittaker, G.W. Automatic calibration of hydrologic models with multi-objective evolutionary algorithm and Pareto optimization. *J. Am. Water Resour. Assoc.* **2007**, *43*, 981–989. [[CrossRef](#)]
11. Muleta, M.K.; Nicklow, J.W. Sensitivity and uncertainty analysis coupled with automatic calibration for a distributed watershed model. *J. Hydrol.* **2005**, *306*, 127–145. [[CrossRef](#)]
12. Tigkas, D.; Christelis, V.; Tsakiris, G. Comparative study of evolutionary algorithms for the automatic calibration of the Medbasin-D conceptual hydrological model. *Environ. Process.* **2016**, *3*, 629–644. [[CrossRef](#)]
13. Afshar, A.; Kazemi, H.; Saadatpour, M. Particle swarm optimization for automatic calibration of large scale water quality model (CE-QUAL-W2): Application to Karkheh Reservoir, Iran. *Water Resour. Manag.* **2011**, *25*, 2613–2632. [[CrossRef](#)]
14. Goktas, R.K.; Aksoy, A. Calibration and verification of QUAL2E using genetic algorithm optimization. *J. Water Res. Plan. Manag.* **2007**, *133*, 126–136. [[CrossRef](#)]
15. Liu, Y.; Khu, S.T. Automatic calibration of numerical models using fast optimisation by fitness approximation. In Proceedings of the International Joint Conference on Neural Networks, Orlando, FL, USA, 12–17 August 2007; pp. 1073–1078.
16. Huang, Y.T.; Liu, L. Multiobjective water quality model calibration using a hybrid genetic algorithm and neural network-based approach. *J. Environ. Eng.* **2010**, *136*, 1020–1031. [[CrossRef](#)]
17. Huang, Y. Multi-objective calibration of a reservoir water quality model in aggregation and non-dominated sorting approaches. *J. Hydrol.* **2014**, *510*, 280–292. [[CrossRef](#)]
18. Afshar, A.; Shojaei, N.; Sagharjooghifarahani, M. Multiobjective calibration of reservoir water quality modeling using Multiobjective Particle Swarm Optimization (MOPSO). *Water Resour. Manag.* **2013**, *27*, 1931–1947. [[CrossRef](#)]
19. Vrugt, J.A.; Gupta, H.V.; Bastidas, L.A.; Bouten, W.; Sorooshian, S. Effective and efficient algorithm for multiobjective optimization of hydrologic models. *Water Resour. Res.* **2003**, *39*, 1214. [[CrossRef](#)]
20. Madsen, H. Automatic calibration of a conceptual rainfall-runoff model using multiple objectives. *J. Hydrol.* **2000**, *235*, 276–288. [[CrossRef](#)]
21. Coello, C.A.C.; Lamont, G.B.; Van Veldhuizen, D.A. *Evolutionary Algorithms for Solving Multi-Objective Problems*; Springer: New York, NY, USA, 2007; pp. 5–35.
22. Deb, K. *Multi-Objective Optimization Using Evolutionary Algorithms*; Wiley: New York, NY, USA, 2001; pp. 13–49.
23. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evolut. Comput.* **2002**, *6*, 182–197. [[CrossRef](#)]
24. Schaffer, J.D. Multiple objective optimization with vector evaluated genetic algorithms. In Proceedings of the First International Conference on Genetic Algorithms and Their Applications, Pittsburg, PA, USA, 24–26 July 1985; Lawrence Erlbaum Associates, Inc.: Hillsdale, NJ, USA, 1985.
25. Konak, A.; Coit, D.W.; Smith, A.E. Multi-objective optimization using genetic algorithms: A tutorial. *Reliab. Eng. Syst. Saf.* **2006**, *91*, 992–1007. [[CrossRef](#)]
26. Xue, F.; Sanderson, A.C.; Graves, R.J. Multi-objective differential evolution and its application to enterprise planning. In Proceedings of the IEEE International Conference on Robotics and Automation, Taipei, Taiwan, 14–19 September 2003; IEEE: Piscataway, NJ, USA, 2003; pp. 3535–3541.
27. Kotinis, M. Improving a multi-objective differential evolution optimizer using fuzzy adaptation and K-medoids clustering. *Soft Comput.* **2014**, *18*, 757–771. [[CrossRef](#)]
28. Coello, C.A.C.; Pulido, G.T.; Lechuga, M.S. Handling multiple objectives with particle swarm optimization. *IEEE Trans. Evolut. Comput.* **2004**, *8*, 256–279. [[CrossRef](#)]
29. Mousa, A.A.; El-Shorbagy, M.A.; Abd-El-Wahed, W.F. Local search based hybrid particle swarm optimization algorithm for multiobjective optimization. *Swarm Evolut. Comput.* **2012**, *3*, 1–14. [[CrossRef](#)]

30. Kennedy, J.; Eberhart, R.C. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; IEEE: Piscataway, NJ, USA, 1995; pp. 1942–1948.
31. Hassanien, A.E.; Emary, E. *Swarm Intelligence: Principles, Advances, and Applications*; CRC Press: New York, NY, USA, 2016; pp. 113–132.
32. Merkle, D.; Middendorf, M. *Swarm Intelligence*; Springer: New York, NY, USA, 2014; pp. 35–71.
33. Margarita, R.-S.; Coello, C.A. Multi-objective particle swarm optimizers: A Survey of the State-of-the-Art. *Int. J. Comput. Intell. Syst.* **2006**, *2*, 287–308.
34. Tsai, S.J.; Sun, T.Y.; Liu, C.C.; Hsieh, S.T.; Wu, W.C.; Chiu, S.Y. An improved multi-objective particle swarm optimizer for multi-objective problems. *Expert Syst. Appl.* **2010**, *37*, 5872–5886. [[CrossRef](#)]
35. Mostaghim, S.; Teich, J. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In Proceedings of the Swarm Intelligence Symposium, Indianapolis, IN, USA, 26–26 April 2003; IEEE Press: Piscataway, NJ, USA, 2003; pp. 26–33.
36. Gong, D.W.; Zhang, Y.; Zhang, J.H. Multi-objective Particle Swarm Optimization Based on Minimal Particle Angle. In *Advances in Intelligent Computing, Proceedings of the International Conference on Intelligent Computing, Hefei, China, 23–26 August 2005*; Springer: New York, NY, USA, 2005; pp. 571–580.
37. Salazar-Lechuga, M.; Rowe, J.E. Particle swarm optimization and fitness sharing to solve multi-objective optimization problems. In Proceedings of the 2005 IEEE Congress on Evolutionary Computation, Edinburgh, Scotland, UK, 2–5 September 2005; IEEE: Piscataway, NJ, USA, 2005; pp. 1204–1211.
38. Agrawal, S.; Panigrahi, B.; Tiwari, M.K. Multiobjective particle swarm algorithm with fuzzy clustering for electrical power dispatch. *IEEE Evolut. Comput.* **2008**, *12*, 529–541. [[CrossRef](#)]
39. Raquel, C.R.; Naval, P.C., Jr. An effective use of crowding distance in multiobjective particle swarm optimization. In Proceedings of the 7th Annual Conference on Genetic and Evolutionary Computation, Washington, DC, USA, 25–29 June 2005; ACM: New York, NY, USA, 2005; pp. 257–264.
40. Tripathi, P.K.; Bandyopadhyay, S.; Pal, S.K. Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients. *Inf. Sci.* **2007**, *177*, 5033–5049. [[CrossRef](#)]
41. Han, H.G.; Lu, W.; Qiao, J.F. An adaptive multiobjective particle swarm optimization based on multiple adaptive methods. *IEEE Trans. Cybern.* **2017**, *47*, 2754–2767. [[CrossRef](#)] [[PubMed](#)]
42. Hu, X.; Eberhart, R. Multiobjective optimization using dynamic neighborhood particle swarm optimization. In Proceedings of the 2002 Congress on Evolutionary Computation, Honolulu, HI, USA, 12–17 May 2002; IEEE: Piscataway, NJ, USA, 2002; pp. 1677–1681.
43. Martínez, S.Z.; Coello Coello, C.A. A multi-objective particle swarm optimizer based on decomposition. In Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation, Dublin, Ireland, 12–16 July 2011; ACM: New York, NY, USA, 2011; pp. 69–76.
44. Moubayed, N.A.; Petrovski, A.; McCall, J. A novel smart multi-objective particle swarm optimisation using decomposition. In Proceedings of the International Conference on Parallel Problem Solving From Nature, Kraków, Poland, 11–15 September 2010; Springer: New York, NY, USA, 2010; pp. 1–10.
45. Zhang, Y.; Gong, D.W.; Ding, Z.H. Handling multi-objective optimization problems with a multi-swarm cooperative particle swarm optimizer. *Expert Syst. Appl.* **2011**, *38*, 13933–13941. [[CrossRef](#)]
46. Laumanns, M.; Thiele, L.; Deb, K.; Zitzler, E. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolut. Comput.* **2002**, *10*, 263–282. [[CrossRef](#)] [[PubMed](#)]
47. Sierra, M.R.; Coello, C.A. Improving PSO-based multi-objective optimization using crowding, mutation and  $\epsilon$ -dominance. In Proceedings of the Evolutionary Multi-Criterion Optimization, Guanajuato, Mexico, 9–11 March 2005; Springer: New York, NY, USA, 2005; pp. 505–519.
48. Hu, W.; Yen, G.G. Adaptive multiobjective particle swarm optimization based on parallel cell coordinate system. *IEEE Trans. Evolut. Comput.* **2015**, *19*, 1–18. [[CrossRef](#)]
49. Wang, Y.J.; Yang, Y.P. Particle swarm optimization with preference order ranking for multi-objective optimization. *Inf. Sci.* **2009**, *179*, 1944–1959. [[CrossRef](#)]
50. Leong, W.; Yen, G.G. PSO-based multiobjective optimization with dynamic population size and adaptive local archives. *IEEE Trans. Syst. Man Cybern. Part B* **2008**, *38*, 1270–1293. [[CrossRef](#)] [[PubMed](#)]
51. Daneshyari, M.; Yen, G.G. Cultural-based multiobjective particle swarm optimization. *IEEE Trans. Syst. Man Cybern. Part B* **2011**, *41*, 553–567. [[CrossRef](#)] [[PubMed](#)]

52. Rahimi-Vahed, A.; Mirghorbani, S.; Rabbani, M. A new particle swarm algorithm for a multi-objective mixed-model assembly line sequencing problem. *Soft Comput.* **2007**, *11*, 997–1012. [[CrossRef](#)]
53. Zheng, L.M.; Wang, Q.; Zhang, S.X.; Zheng, S.Y. Population recombination strategies for multi-objective particle swarm optimization. *Soft Comput.* **2016**, *21*, 4693–4705. [[CrossRef](#)]
54. Ercan, M.B.; Goodall, J.L. Design and implementation of a general software library for using NSGA-II with SWAT for multi-objective model calibration. *Environ. Model. Softw.* **2016**, *84*, 112–120. [[CrossRef](#)]
55. Zhang, X.S.; Beeson, P.; Link, R.; Manowitz, D.; Izaurralde, R.C.; Sadeghi, A.; Thomson, A.M.; Sahajpal, R.; Srinivasan, R.; Arnold, J.G. Efficient multi-objective calibration of a computationally intensive hydrologic model with parallel computing software in Python. *Environ. Model. Softw.* **2013**, *46*, 208–218. [[CrossRef](#)]
56. Sun, J.W.; Lu, C.; Wang, M.X.; Yuan, H.; Qi, L. Performance assessment and prediction for superheterodyne receivers based on Mahalanobis distance and time sequence analysis. *Int. J. Antennas Propag.* **2017**, *2017*, 6458954. [[CrossRef](#)]
57. Wang, Z.P.; Lu, C.; Wang, Z.L.; Liu, H.; Fan, H. Fault diagnosis and health assessment for bearings using the Mahalanobis-Taguchi system based on EMD-SVD. *Trans. Inst. Meas. Control* **2013**, *35*, 798–807. [[CrossRef](#)]
58. Xing, E.P.; Ng, A.Y.; Jordan, M.I.; Russell, S. Distance metric learning, with application to clustering with side-information. In Proceedings of the Advances in Neural Information Processing Systems, Cambridge, MA, USA, 2003; MIT Press: Cambridge, MA, USA, 2003; pp. 521–528.
59. Hoi, S.C.H.; Liu, W.; Chang, S.F. Semi-supervised distance metric learning for collaborative image retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 1–7.
60. Haldar, N.A.H.; Khan, F.A.; Ali, A.; Abbas, H. Arrhythmia classification using Mahalanobis distance based improved fuzzy C-means clustering for mobilehealth monitoring systems. *Neurocomputing* **2016**, *220*, 221–235. [[CrossRef](#)]
61. Bhamre, T.; Zhao, Z.; Singer, A. Mahalanobis distance for class averaging of cryo-EM images. In Proceedings of the IEEE International Symposium on Biomedical Imaging, Melbourne, Australia, 18–21 April 2017; pp. 654–658.
62. Du, X.J.; Shao, F.J.; Wu, S.Y.; Zhang, H.L.; Xu, S. Water quality assessment with hierarchical cluster analysis based on Mahalanobis distance. *Environ. Monit. Assess.* **2017**, *189*, 335. [[CrossRef](#)] [[PubMed](#)]
63. Xiang, S.M.; Nie, F.P.; Zhang, C.S. Learning a Mahalanobis distance metric for data clustering and classification. *Pattern Recognit.* **2008**, *41*, 3600–3612. [[CrossRef](#)]
64. Farber, O.; Kadmon, R. Assessment of alternative approaches for bioclimatic modeling with special emphasis on the Mahalanobis distance. *Ecol. Model.* **2003**, *160*, 115–130. [[CrossRef](#)]
65. McLachlan, G.J. Mahalanobis distance. *Resonance* **1999**, *4*, 20–26. [[CrossRef](#)]
66. Zitzler, E.; Deb, K.; Thiele, L. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolut. Comput.* **2000**, *8*, 173–195. [[CrossRef](#)] [[PubMed](#)]
67. Deb, K.; Thiele, L.; Laumanns, M.; Zitzler, E. Scalable multi-objective optimization test problems. In Proceedings of the 2002 Congress on Evolutionary Computation, Honolulu, HI, USA, 12–17 May 2002; IEEE: Piscataway, NJ, USA, 2002; pp. 825–830.
68. Van Veldhuizen, D.A.; Lamont, G.B. On measuring multiobjective evolutionary algorithm performance. In Proceedings of the 2000 Congress on Evolutionary Computation, La Jolla, CA, USA, 16–19 July 2000; IEEE: Piscataway, NJ, USA, 2000; pp. 204–211.
69. Schott, J.R. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.
70. Hua, Z. *The Foundation of Environmental Hydrodynamic*; Science Press: Beijing, China, 2016; pp. 102–121.
71. Cerco, C.F.; Cole, T. *User's Guide to the CE-QUAL-ICM Three-Dimensional Eutrophication Model: Release Version 1.0*; US Army Engineer Waterways Experiment Station: Vicksburg, MS, USA, 1995.
72. Cerco, C.F.; Noel, M.R. Twenty-one-year simulation of Chesapeake Bay water quality using the CE-QUAL-ICM eutrophication model. *J. Am. Water Resour. Assoc.* **2013**, *49*, 1119–1133. [[CrossRef](#)]
73. Li, X.; Zhang, S.; Wang, L.; Gou, M.; Xu, X. Coupling the EFDC and CE-QUAL-ICM models to simulate water quality of shallow lake in Inner Mongolia, China. In Proceedings of the 2015 International Conference on Sustainable Development (ICSD2015), Wuhan, China, 25–27 September 2016; World Scientific Publishing Company: Singapore, 2016; pp. 888–897.

74. Yang, L.B.; Lei, K.; Meng, W.; Fu, G.; Yan, W.J. Temporal and spatial changes in nutrients and chlorophyll- $\alpha$  in a shallow lake, Lake Chaohu, China: An 11-year investigation. *J. Environ. Sci.* **2013**, *25*, 1117–1123. [[CrossRef](#)]
75. Xie, P. *Exploring the History of Chaohu Lake: About Cyanobacteria, Eutrophication and Geological Evolution*; Science Press: Beijing, China, 2009; pp. 132–140.
76. Pu, P.M.; Tu, Q.Y.; Wang, S.M. Progress of limnology in China. *Chin. J. Oceanol. Limnol.* **1991**, *9*, 193–206. [[CrossRef](#)]
77. Jørgensen, S.E.; Bendricchio, G. *Fundamentals of Ecological Modelling*; Elsevier: New York, NY, USA, 2001; pp. 222–255.
78. Xu, F.L.; Jørgensen, S.E.; Tao, S.; Li, B.G. Modeling the effects of ecological engineering on ecosystem health of a shallow eutrophic Chinese lake (Lake Chao). *Ecol. Model.* **1999**, *117*, 239–260. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).