

Article

# Optimizing Inverse Distance Weighting with Particle Swarm Optimization

Alina Barbulescu <sup>1,\*</sup>, Andrei Bautu <sup>2</sup> and Elena Bautu <sup>1</sup> 

<sup>1</sup> Department of Mathematics and Computer Science, Ovidius University of Constanta, 900527 Constanta, Romania; ebautu@univ-ovidius.ro

<sup>2</sup> Department of Navigation and Naval Transport, Mircea cel Batran Naval Academy, 900218 Constanta, Romania; abautu@anmb.ro

\* Correspondence: alinadumitriu@yahoo.com

Received: 21 January 2020; Accepted: 11 March 2020; Published: 18 March 2020



**Abstract:** Spatial analysis of hydrological data often requires the interpolation of a variable from point samples. Commonly used methods for solving this problem include Inverse Distance Weighting (IDW) and Kriging (KG). IDW is easily extensible, has a competitive computational cost with respect to KG, hence it is usually preferred for this task. This paper proposes the optimization of finding the IDW parameter using a nature-inspired metaheuristic, namely Particle Swarm Optimization (PSO). The performance of the improved algorithm is evaluated in a complex scenario and benchmarked against the KG algorithm for 51 precipitation series from the Dobrogea region (Romania). Apart from facilitating the process of applying IDW, the PSO implementation for Optimizing IDW (OIDW) is computationally lighter than the traditional IDW approach. Compared to Kriging, OIDW is straightforward to be implemented and does not require the difficult process of identification of the most appropriate variogram for the given data.

**Keywords:** inverse distance weighting; kriging; particle swarm optimization; prediction error; spatial interpolation

## 1. Introduction

Precipitation data analysis plays a major role in the process of making informed decisions for water resources management. Such data are gathered at monitoring stations scattered throughout a region. They are either not readily available for other locations or may be missing at certain time periods. These issues have negative influences on hydrological studies that rely on precipitation data as input. Therefore, spatial interpolation methods prove to be useful for estimating precipitation data at unsampled locations [1].

Spatial interpolation methods are commonly used for the prediction of the values of environmental variables. Their specificity comes from incorporating information related to the geographic position of the sample data points. The most used methods are usually classified in the categories [2,3]: (1) deterministic methods (Nearest Neighbor, Inverse Distance Weighting (IDW), Splines, Classification and Regression methods), (2) geostatistical methods (Kriging (KG)) or (3) combined methods (regression combined with other interpolation methods and classification combined with other interpolation methods).

The importance of a successful spatial interpolation of precipitation before the hydrological modeling and a thorough review of different spatial interpolation methods for its modeling are presented in [1]. Among them, IDW stands out as a real competitor. It builds estimates for environmental variables at a location, based on the values of those variables at some nearby sites and on the distances between that location and those where the variable is known.

The interpolation method often chosen by geoscientists is IDW, implemented in many Geographic Information Systems (GIS) packages [4]. Its popularity is mainly due to its straightforward interpretability, easy computability and good prediction results. Still, the No Free Lunch theorem for optimization [5] stands also in the case of spatial interpolation: no method stands out as being the best in all situations [6]. In a study about the rainfall data in the Indian Himalayas, Kumari et al. [7] performed a comparison of several interpolation methods, among which there were several variants of multivariate Kriging and IDW. It was shown that none of the methods performs best in all the studied cases. Principal Component Regression with the Residual correction (PCRR) method, IDW and the Multiple Linear Regression (MLR) methods were used to compare interpolated annual, daily and hourly precipitation and the spatial distribution of precipitation in the Xinxie catchment [8], while the spatial distribution of precipitation deficit over Seyhan River basin using IDW is reported in [9].

The basic IDW method is successfully employed in [10] to estimate the rainfall distribution in a region of Iraq. The scenario uses incremental values for the power parameter, in the range from 1 to 5. A modified IDW method is investigated in [11], where the elevation is also considered for estimating the values at unknown locations. A new method for estimating the regional precipitation (MPPM) has been introduced in [12] and its performance has been tested against that of IDW and kriging. It was reported that MPPM avoids the problems that could appear in the application of kriging methods, as (i) the invertibility of the distance matrix, (ii) the high computational cost related to building the inverse of the distance matrix in the case of a high number of stations, (iii) the choice of the model for the estimation of theoretical variogram and (iv) the selection of the optimal parameters of the variogram model. It was shown that deterministic methods could be good competitors for spatial interpolation techniques, such as KG, performing better than the last ones in the study cases presented in [12]. In an engineering setting, Gholipour et al. [13] investigate a hybridization of IDW with a harmony search. The obtained metamodel significantly reduces the computational effort and improves the convergence rate. A genetic algorithm is used in [6] to find the optimal order of distances in IDW.

An adaptive version of the IDW method is introduced in [4], where the authors suggest that the spatial pattern of the sampled stations in the neighborhood could influence the weighting parameter. The algorithm selects the weights in IDW based on the stations' density around the unsampled location. It gave better estimations than KG on the study cases. However, the choices of the membership functions and the number of relevant neighbors are heuristics, and there is no guarantee that they will perform well in other scenarios.

Some machine learning models were also investigated in [3], with application to spatial interpolation of environmental variables. They are compared with several traditional spatial interpolation methods, among which is inverse distance squared. The study found that the combination of random forest and IDW, respectively random forest with OK were the best methods, with similar accuracies. At the same time, OK was found to perform similarly to IDW in the tested scenarios. A hybrid method that combines IDW with support vector machines is reported in [14], applied to multiyear annual precipitation. In [15], temporal predictions obtained with an ensemble approach are used as inputs to spatial interpolation by IDW, with improved overall results.

The main difficulty faced when applying IDW is setting the value of the power parameter. This is usually done before the algorithm is applied. The usual approach for searching for the optimal value of the power parameter is by exhaustive search, by sampling all possible values in a given interval, at a user chosen step size [2]. The results of this method depend on the search window. The method is only guaranteed to find a local optimum [2].

In this paper, we target the optimization of the process of setting the power parameter in IDW, using a nature-inspired metaheuristic—Particle Swarm Optimization. We automate the process of identifying the suitable parameter, while maintaining, if not optimizing, the prediction accuracy of the standard IDW method. Experiments are performed on maximum annual precipitation data gathered in the Dobrogea region (Romania). An evaluation of the proposed hybrid method is performed, against standard IDW and Kriging.

The paper structure is the following. Section 2 begins with a brief presentation of the meteorological stations locations in the study area. Sections 2.2 and 2.3 present the spatial interpolation methods IDW and KG. The Particle Swarm Optimization (PSO) metaheuristic is presented in Section 2.4. The section ends with the presentation of the hybrid algorithm Optimizing IDW (OIDW), obtained by optimizing the  $\beta$  parameter of IDW with PSO. The experimental settings are described in Section 3, while the results are discussed in Section 4. The paper ends with the conclusion section.

## 2. Materials and Methods

### 2.1. Study Area

The studied area is Dobrogea, a region of 15,500 km<sup>2</sup> located in the South-East of Romania, between the Danube River and the Black Sea, between 27°15'15"–29°30'10" eastern longitude and 43°40'4"–45°25'3" northern latitude [16]. Its geographic structure is that of a plateau with a hilly aspect, the altitude decreasing from north to south. The climate is temperate–continental, but the region is subject to drought and desertification [16,17].

The study data consists of 51 series of 1-day maximum annual precipitation registered during the period January 1965–December 2005 at 10 main meteorological stations and 41 hydro-meteorological points in the region of Dobrogea, Romania (Figure 1). When working only with the data recorded at the main stations, the input data will be a table with 41 rows and 10 columns (in the A1 and B1 scenarios presented in Section 3), while when working with all data, the input will consist of a table with 41 rows and 51 columns (in the A2 and B2 scenarios presented in Section 3). The data were collected at the stations under the National Authority Romanian Waters; they are without gaps and they are reliable. The maximum annual series collected at the main meteorological stations are represented in Figure 2.



Figure 1. The map of the Dobrogea region and the stations where the data series were recorded.

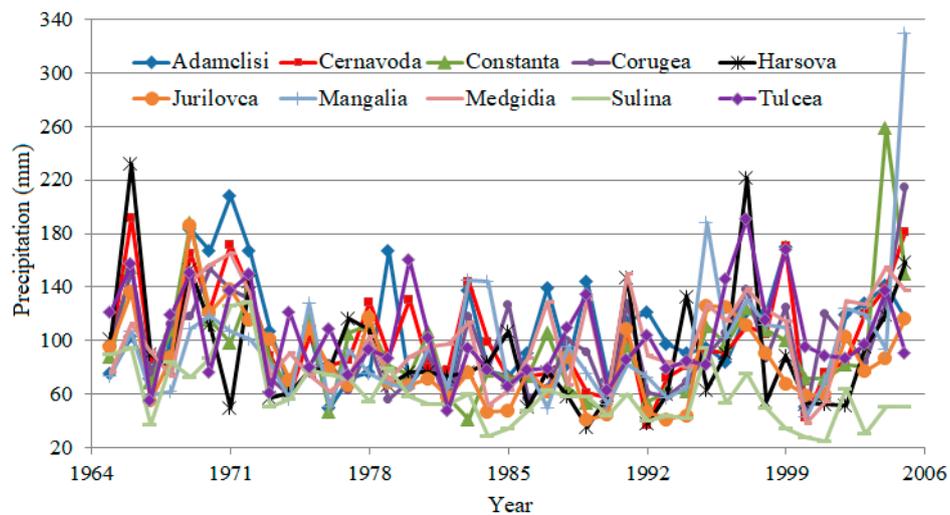


Figure 2. The maxima series recorded at the main stations.

### 2.2. IDW Interpolation

Essentially, all spatial interpolation methods compute weighted averages of sampled data, as estimations for unknown data [2,3]. Given a set of spatial data of an attribute  $z$  at  $n$  locations, the general estimation formula is:

$$z(\hat{s}_0) = \sum_{i=1}^n w_i(s_0) \cdot z(s_i) \tag{1}$$

where  $z(\hat{s}_0)$  is the interpolated value estimated for the variable of interest at the station  $s_0$ ,  $z(s_i)$  is the sample value at the station  $s_i$ ,  $w_i(s_0)$  is the weight attached to the station  $s_i$  and  $n$  is the stations' number.

The main difference between all spatial interpolation methods relies in computing the weights  $w_i$  used in the interpolation.

The problem we tackled consists of estimating precipitation values for some locations where these values are unknown, using as input data the precipitation values recorded at several locations in the neighborhood. The general Formula (1) is also valid for IDW. The simplest version of weights estimation uses the inverse distances from all the points to the target one [18]:

$$w_i(s_0) = \frac{1/d(s_0, s_i)^\beta}{\sum_{i=1}^n (1/d(s_0, s_i)^\beta)}, \beta > 1, \tag{2}$$

where  $d(s_0, s_i)$  is the distance from  $s_0$  to  $s_i$  and  $\beta$  is a parameter that must be determined. Thus, the weights decrease as the distance increases, especially if the value of  $\beta$  is large. The parameter  $\beta$  determines the degree of influence the neighboring stations have upon the estimates for a given station (it is expected that nearer neighbors have more influence upon the estimated value than the more distant ones).

Choosing  $\beta$  is an optimization process by itself. Usually, the search for the optimal  $\beta$  is a grid search: a specific range is set (arbitrarily or based on some intuition of the researcher), and then  $\beta$  takes all values in that range, with a certain step-size, also arbitrarily chosen. The value yielding the lowest prediction error (among the searched values) is finally attributed to the parameter.

### 2.3. Kriging

Kriging (KG) is the generic name given to a family of generalized least-squares regression algorithms, used in geostatistics for data spatial interpolation. The spatial correlation was analyzed in

geostatistics using the variogram. The main ideas behind the methodology for Kriging are shortly presented below, based upon the bibliographic resources [19,20].

Given a random function  $Z(s)$  that is stationary, with a constant mean  $E(Z(s)) = \mu$ , the semivariogram is defined by:

$$\gamma(h) = \frac{1}{2}E(Z(s_i) - Z(s_i + h))^2, \tag{3}$$

where  $Z(s_i)$  and  $Z(s_i + h)$  are the variable values at the study station and at a location situated at the distance  $h$  from the study location. When assuming the direction independence of the semivariance (isotropy), the variogram can be estimated using  $\Sigma$  the sample variogram, defined by

$$\hat{\gamma}(h) = \frac{1}{2N_h} \sum_{i=1}^{N_h} (Z(s_i) - Z(s_i + h))^2 \tag{4}$$

where  $N_h$  is the number of sample pairs  $(Z(s_i), Z(s_i + h))$  used in estimation and where  $N(h)$  is the number of data pairs, which are approximately separated by the lag  $h$ .

After computing the sample variogram, one should determine the empirical semivariogram by fitting a parametric model to it. This can be done by the Generalized Least Squares (GLS), Maximum Likelihood or Bayesian methods [19]. GLS is the method used in this article.

The nugget, sill and range are the parameters used to describe a variogram (Figure 3). The nugget is the random error process, shown by the height of the jump of the semivariogram at the discontinuity at the origin. The sill is the variogram limit, when the lag tends to infinity. The range is the minimum lag at which the difference between the variogram and sill becomes negligible [21].

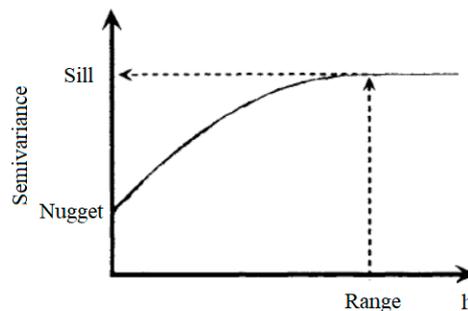


Figure 3. A variogram and its parameters [21].

Different types of variograms may be used, as function of the data series, for example spherical, exponential, Gaussian, Matern and power [19].

At the final modeling stage of ordinary KG, the predictions are based on the model:

$$Z(s) = \mu + \varepsilon(s) \tag{5}$$

where  $\mu$  is the mean and  $\varepsilon(s)$  is the spatially correlated stochastic part of variation. The predictions are obtained by formula:

$$\hat{z}_0(s_0) = \sum_{i=1}^n w_i(s_0)z(s_i) = \lambda_0^T z \tag{6}$$

where  $\lambda_0^T$  is the transposed KG weights vector ( $w_i$ ), and  $z$  is the vector containing the observations at  $n$  neighbor locations.

The Kriging predictions are obtained by:

$$\hat{z}_0(s_0) = \sum_{i=1}^n w_i(s_0)z(s_i) = \mathbf{w}_0^T z \tag{7}$$

where  $w_0^T$  is the transposed kriging weights vector ( $w_i$ ), and  $z$  is the vector containing the observations at  $n$  neighbor locations. The weights are based on the covariances among points in the sample and the covariances between sample points and the point to be predicted. The Kriging estimator should be unbiased and the error variance is minimized.

The weights for ordinary Kriging can be found by solving the Kriging equations:

$$w = \begin{bmatrix} C_{11} & \dots & C_{1n} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ C_{n1} & \dots & C_{nn} & 1 \\ 1 & \dots & 1 & 0 \end{bmatrix}^{-1} \begin{bmatrix} C_{10} \\ \vdots \\ C_{n0} \\ 1 \end{bmatrix} \tag{8}$$

where  $C_{ij} = \text{Cov}(Z_i, Z_j)$ ,  $C_{i0} = \text{Cov}(Z_i, Z_0)$  and  $\lambda$  is the Lagrange multiplier that appears due to the constraint  $\sum_{i=1}^n w_i = 1$ .

Since the first step of the KG procedure is building a variogram (not a covariogram), one should want to determine the Kriging equation, in terms of variogram. Under the hypothesis of second-order stationarity:

$$C_{ij} = \sigma^2 - \gamma_{ij} \tag{9}$$

$\sigma^2$  is the variance and  $\gamma_{ij}$  is the semivariance. Therefore, (8) can be written in the equivalent form:

$$\begin{bmatrix} -\gamma_{11} & \dots & -\gamma_{1n} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ -\gamma_{n1} & \dots & -\gamma_{nn} & 1 \\ 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_n \\ \lambda \end{bmatrix} = \begin{bmatrix} -\gamma_{10} \\ \vdots \\ -\gamma_{n0} \\ 1 \end{bmatrix} \tag{10}$$

The generalized least squared (GLS) estimate of the global mean of the data in the study region is given by:

$$\hat{m}_{GLS} = (\mathbf{1}^T \mathbf{C}_1^{-1} \mathbf{1})^{-1} \mathbf{1}^T \mathbf{C}_1^{-1} \mathbf{z} \tag{11}$$

where  $\mathbf{1} = (1, \dots, 1)$  and  $\mathbf{C}_1 = \begin{bmatrix} C_{11} & \dots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \dots & C_{nn} \end{bmatrix}$  [22].

The KG goodness of fit is influenced by the spatial data structure, the variogram choice and the number of data points chosen for the computation [23]. KG is more reliable when the number of stations is big enough [24]. For details on different approaches to Kriging study, the readers may refer to [19,20,25].

#### 2.4. Particle Swarm Optimization (PSO)

Metaheuristics are problem-independent (stochastic optimization) techniques [26]. They are strategies that guide the search process, for an efficient exploration of the search space, for finding near-optimal solutions. Metaheuristic algorithms are approximate and usually non-deterministic, and they usually find “good” solutions in a “reasonable” amount of time [27]. Their advantage is that they are noise tolerant; they do not need the source code of the evaluation simulation, which can be used as a black box.

Swarm Intelligence (SI) comprises a class of novel population based intelligent metaheuristics, inspired from the emergent intelligent behavior of flocks of birds, fish, ants, immune system, bacteria, etc. In such collectivities, social interactions lead to complex social behavior, and that behavior is modified in dynamic environments. Particle Swarm Optimization (PSO) represents a branch of algorithms of SI, which implements a computational model for social learning [28]. The algorithm works with a population of particles (swarm) that, initially, are randomly generated throughout the

search space of the problem. The particles are evaluated against a fitness function, each of them searching for locations in the fitness landscape with better fitness values. Each particle has neighbors, with which it exchanges information, and which are dictated by the representation of the particle and the neighborhood topology.

The PSO algorithm consists in a swarm of particles that move in a (multidimensional) search space. Each particle has a position  $x$ , a speed  $v$ , a memory of its most recent success (pbest—personal best—the position where the particle has obtained the best fitness) and a memory of the best neighbor (gbest—global best—the particle with the best neighbor in the neighborhood). The particles' speed and positions are updated after each iteration, according to the following equations:

$$v[t + 1] = w_1 v[t] + w_2 \text{rand}() + w_3 \text{rand}(g\text{best} - x) \quad (12)$$

$$x[t + 1] = x[t] + v[t + 1] \quad (13)$$

The parameters that appear in Equation (5) are:

- $w_1$  the inertia weight, which forces the particle to move in the same direction; it balances exploration and exploitation. When  $w_1$  is high, PSO is focused on the search space's exploration. When  $w_1$  is small, PSO focuses on exploitation rather than on exploration. Scenarios in which  $w_1$  decreases in time are usually used;
- $w_2$  and  $w_3$  the learning factors—they are weights of the acceleration that attracts the particle towards its personal best position or the global best position.  $w_2$  is the cognitive learning factor, that suggests the tendency to repeat personal actions that proved more successful, while  $w_3$  is the social learning factor—a measure of the tendency to follow the success of the best individual from its neighborhood.

After updating the speed, a speed limitation rule is usually applied, to prevent the particles from moving chaotically in the search space:

$$v[t + 1] = \max(-v_{\max}, \min(v_{\max}, v[t + 1])) \quad (14)$$

where  $v_{\max}$  is a parameter of the algorithm.

The basic structure of the PSO Algorithm 1 is the following:

---

#### Algorithm 1

---

```

1: t = 0;
2:   Create the initial swarm P(0)
3:   repeat
4:     Evaluate particles in P(t) using the fitness function
5:     Update pbest for each particle
6:     Update gbest
7:     t = t + 1;
8:     foreach particle in P(t)
9:       Update its speed using Equation (12)
10:      Update its speed using Equation (14)
11:      Update its position using Equation (13)
12:     end foreach
13:   until stopping criterion is met

```

---

#### 2.5. Optimization of the IDW Parameter with PSO

The optimization we propose is to identify the value of the optimal  $\beta$ , from (2) using PSO as the search algorithm. The PSO metaheuristic was chosen due to its simplicity with respect to coding,

small computational cost and reduced number of parameters (by comparison with genetic algorithms, for example). PSO's documented success in optimization applications in a wide range of engineering problems [28,29] was also a decisive factor in choosing PSO over other metaheuristics.

The particles' positions in the PSO algorithm encode candidate solutions for the power parameter in Formula (2) of the IDW weights. They are randomly generated initially within the interval [1.0001, 5] (since  $\beta > 1$ , from the IDW definition).

The fitness function used in the algorithm assigns to each particle the mean squared error (MSE) of the predictions made with IDW, using as weight the power parameter  $\beta$  encoded by the respective particle:

$$\text{fitness}(\beta) = \frac{1}{n} \sum_{i=1}^n (p_i - m_i)^2 \tag{15}$$

where:

- $n$  is number of stations used in the IDW,
- $p_i$  is the value predicted by IDW for the precipitation at station  $i$ ,
- $m_i$  is the (known) measured value for the precipitation at station  $i$ .

As far as we know, the proposed approach is new in the literature.

In the following we will refer to the improved algorithm as Optimized IDW (OIDW). Figure 4 presents the flowchart of the OIDW algorithm.

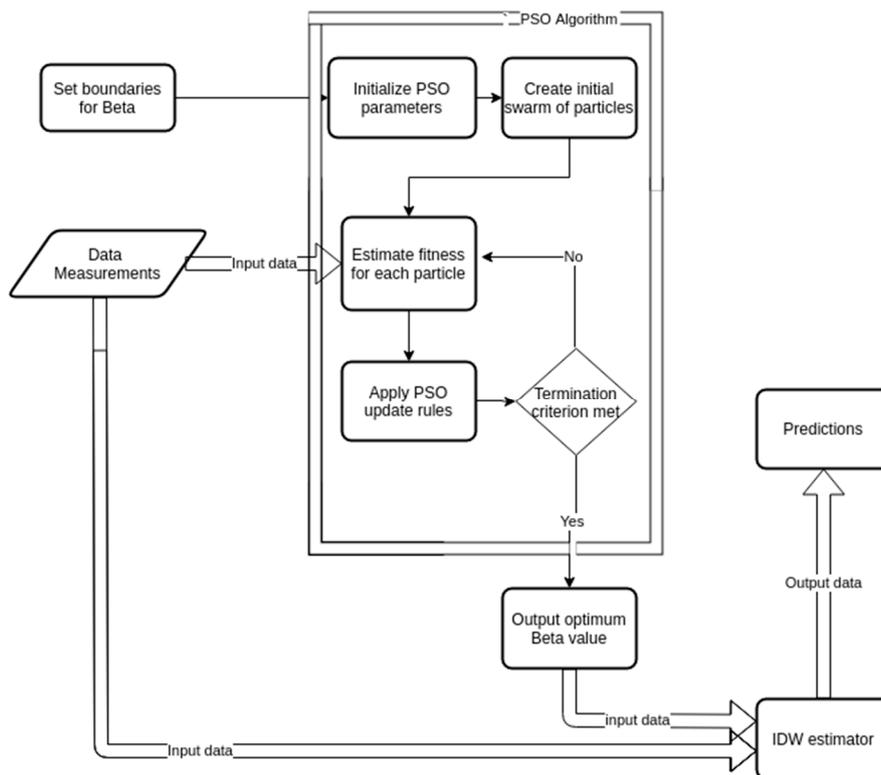


Figure 4. The flowchart of the Optimizing Inverse Distance Weighting (OIDW) algorithm.

### 3. Experimental Settings

The experiments were run using an improved deterministic PSO variant from [30], which is among the most popular versions of the PSO algorithm [31]. We are interested in assessing the quality of predictions for OIDW and compare them with the traditional grid search IDW (denoted, in the following, by IDW\*).

The experiments were run in 2 scenarios:

(Scenario A)

First, we used PSO and grid search to identify a single  $\beta$  that minimizes the sum of prediction error over all stations. The  $\beta$  value identified is characteristic of the entire Dobrogea region, hence it can be useful to infer values at points where no recordings are available, for example. When the predicted values for a given station are computed, the weights in Formula (1) corresponding to that station are set to 0.

(Scenario B)

We used PSO and grid search to search a  $\beta$  for each station that minimizes the prediction error when the series from that station is estimated using the other series from the other stations.

For example, if we have ten stations, to estimate the precipitation at the station I, the series recorded at the stations II-X are used as input, while the series recorded at the station I are hold out as control values. Thus, they are used for comparison with the values computed by applying Formula (2) with the parameter  $\beta_1$  identified by OIDW. The  $\beta_1$  obtained minimizes the MSE for the station I. Another example: to estimate the values of the series from the station V, the series recorded at station I-IV and VI-X are used as an input, and a  $\beta_5$  is computed, corresponding to the minimum MSE for the station V, and so on.

Further, we refined the experimental settings by performing experiments in two stages. First, we employed a reduced dataset comprising the series from ten main meteorological stations. We will refer to these scenarios as (A1) and (B1). The same experiments were repeated with the extended dataset that contains all 51 series. We will refer to these scenarios as (A2) and (B2). Hence we obtained four scenarios. We took a special interest in type (A) experiments because they should yield a unique empirically determined  $\beta$  to be characteristic for the entire region.

For each scenario, 50 different runs of the PSO algorithm were performed. The default Mersenne twister randomizer with an initial seed of 0 was used in the experiments reported in the tables below.

Setting the PSO parameters to optimal values is a very difficult optimization problem by itself [30,32]. In our experiments, the PSO parameters are chosen to match those widely used in the literature [30]:

- The swarm size is 24,
- Personal best influence is 2,
- Global best influence is 2,
- The inertia weight, decreasing during the iterations of the algorithm from 0.9 to 0.4,
- The number of epochs (i.e., PSO iterations) is 100,
- $\beta$  is searched in the interval (1, 5).

The results are compared with those obtained by applying IDW and KG. For IDW, we identified the parameter  $\beta$  performing a grid search with the step-size of  $10^{-4}$ , in the interval [1.001, 5]. The grid search identifies the parameter  $\beta$  that yields the minimum prediction error (that will be listed in the results tables).

The experiments for OIDW and IDW\* were performed in the Matlab environment, in Matlab R2012, on a computer with Intel Core i5 quad-core processor at 2.30 GHz with 8 GB RAM, using the PSO toolbox, available online [33]. Matlab was also used for performing the IDW\* computation (IDW with the grid search).

For KG, we used Ordinary Kriging and report the mean prediction error obtained. The Kriging algorithm was run using the `gstat`, `sp`, `spacetime`, `automap` and `geoR` libraries from the R software. Different types of variograms (spherical, exponential, Gaussian, linear and power) were fit to the data, using `autofitVariogram` (from `automap` package) [34] and the best one (in terms of the lowest MSE) was selected. While for using the `fit.variogram` function (from `gstat` package) [35] the user has to supply an initial guess (estimate) for the sill, range, etc. to fit a certain type of variogram, the use of the `autofitVariogram` function (from `automap` package) does not necessitate any initial guess.

`autofitVariogram` has the advantage to provide this estimate by computing it based on the data, and then calls the `fit.variogram` function [35]. So, `autofitVariogram` automatically provides fit parameters (sill, nugget and range) of the best variogram. For a deeper insight on the variogram fitting, one can see the vignettes of the mentioned packages on the website of R software [34,35]. In our case, the fit variogram was of an exponential type, with the parameters sill = 144.9766 and range = 48,843.73.

To check if the MSE obtained in OIDW (A and B scenarios) and IDW\* are not statistically different, the following statistical tests have been performed:

- The Anderson–Darling test [36], for testing the null hypothesis of data series normality:

The series is Gaussian (has a normal distribution),

Against the alternative that:

The series is not Gaussian (has not a normal distribution).

- The Levene’s test [36] for verifying the null hypothesis that:

The MSE series (from the scenarios A, B and IDW\*) have the same variance,

Against the alternative that:

The MSE series do not have the same variance;

- The non-parametric Kruskal–Wallis test [37], for checking the null hypothesis that:

The mean ranks of the groups of MSE series (from the scenarios A, B and IDW\*) are the same.

Against the alternative:

The mean ranks of the groups are not the same.

- The non-parametric `eqdist.etest` test [38], for verifying the null hypothesis that:

The MSE series (from the scenarios A, B and IDW\*) have the same distribution,

Against the alternative:

The MSE series did not have the same distribution.

These nonparametric tests were selected because the normality hypothesis was rejected for some MSE series in A, B scenarios and IDW\*. These tests were performed at the significance level of 5%, using the R software, version R 3.5.1.

#### 4. Results and Discussion

The experimental results of the study are presented in Tables 1–4, which contain:

- $\beta$  values rounded up to 4 decimals (which are mean values for the  $\beta$ 's obtained over the 50 runs of the PSO algorithm);
- Standard deviations of the computed  $\beta$  values, denoted by `st.dev.`;
- Mean squared errors (MSE) of the series values obtained in the experiments. The MSE is the average squared difference between the estimated values and the actual values; it is a measure of a model’s quality (Formula (15)). The smaller the MSE, the better the model is.
- The time (in seconds) for running an experiment in the case of IDW\*;
- The average run time (in seconds) over all the 50 experiments and the standard deviation of the time in the OIDW experiments.

PSO is a stochastic algorithm; hence, mean and standard deviations are reported for both the optimized parameter, and the time consumed by the algorithm. The standard deviation is a proof of the algorithm stability (i.e., when the standard deviation is very small it means that the algorithm converges in every run; hence, it is stable and reliable).

The results obtained using only the series recorded at the ten main stations in the experiments done in the scenario A1 are displayed in Table 1, as follows: IDW\* (columns 2–4), OIDW (columns 5–7) and KG (column 8). The values of the  $\beta$  parameter are equal up to the fourth decimal in IDW\* and OIDW. The average run time for IDW\* is 82.19 times greater than that for running OIDW (column 4–60 s and column 7–0.73 s).

Table 2 displays the results of the experiments performed using the series recorded at the main meteorological station, in the scenario B1.

We remarked that the PSO convergence was very good. The standard deviations of time and  $\beta$  in the OIDW algorithm are of the order of  $10^{-4}$ , respectively  $10^{-6}$ , hence they are not reported individually in the table. The values identified by OIDW coincide with those identified by IDW\*, by the 4th decimal. A significant difference appears in the computational time. In the scenario B1, the average computational time for an OIDW run is more than 60 times smaller than the time for a IDW\* run.

**Table 1.** The values of the  $\beta$  parameter obtained in experiments performed using the series recorded at the ten main meteorological stations, in scenario A1. For OIDW, the values in the parenthesis are the standard deviation of the parameter identified, computed over the 50 independent runs. Comparison with KG.

Station	IDW*			OIDW		KG	
	$\beta$	MSE	Time (s)	$\beta$ (std.dev)	MSE	MSE	
Adamclisi	1.1933	32.4184	60	1.1933618 ( $3.2533 \times 10^{-4}$ )	32.4184	0.73 (0.132)	32.73
Cernavoda		22.9189			22.9189		23.40
Constanta		30.3249			30.3249		30.22
Corugea		22.4062			22.4062		22.48
Harsova		35.6281			35.6281		35.03
Jurilovca		24.1018			24.1018		23.04
Mangalia		42.0429			42.0429		42.73
Medgidia		22.3809			22.3809		22.58
Sulina		44.0204			44.0204		43.54
Tulcea		33.0844			33.0844		34.47

**Table 2.** The values of the  $\beta$  parameter obtained in experiments performed using the series recorded at the 10 main meteorological stations, in the scenario B1. Comparison with KG.

Station	IDW*			OIDW		KG	
	$\beta$	MSE	Time (s)	B	MSE	MSE	
Adamclisi	5	28.5774	3.6	5	28.5774	0.059	32.73
Cernavoda	1.0001	22.6682	3.6	1.0001	22.6820	0.049	23.40
Constanta	1.0001	30.1713	3.6	1.0001	30.1713	0.049	30.22
Corugea	1.0001	22.3283	3.6	1.0001	22.3283	0.049	22.48
Harsova	3.0589	35.2566	3.6	3.05951	35.2566	0.056	35.03
Jurilovca	1.0001	23.9813	3.6	1.0001	23.9813	0.05	23.04
Mangalia	1.0001	41.9517	3.6	1.0001	41.9517	0.049	42.73
Medgidia	1.5189	22.3073	3.6	1.5189	22.3073	0.081	22.58
Sulina	2.6220	43.4118	3.6	2.6219	43.4118	0.065	43.54
Tulcea	1.0001	33.0501	3.6	1.0001	33.0501	0.049	34.47

**Table 3.** Parameter  $\beta$  values obtained in experiments performed using the series recorded at all (51) meteorological stations, in the scenario A2. For OIDW, the values in the parentheses are the standard deviation of the parameter identified, computed over the 50 independent runs. Comparison with KG.

Station	IDW*			OIDW		KG	
	$\beta$	MSE	Time (s)	$\beta$ (std.dev)	MSE	MSE	
Adamclisi	1.5921	29.1151	190	1.59207 ( $1.1813 \times 10^{-4}$ )	29.1151	4.846 (1.0498)	31.78
Cernavoda		22.166			22.166		25.78
Constanta		29.2801			29.28		36.47
Corugea		17.9985			17.9984		29.95
Harsova		36.2216			36.2214		31.51
Jurilovca		32.2055			32.2054		37.43
Mangalia		39.0367			39.0366		27.38
Medgidia		29.2706			29.2702		31.47
Sulina		46.5612			46.5612		28.05
Tulcea		24.5771			24.5772		28.94

**Table 4.** Parameter  $\beta$  values obtained in experiments performed using the series recorded at all (51) meteorological stations, in the scenario A2. For OIDW, the values in the parentheses are the standard deviation of the parameter identified, computed over the 50 independent runs. Comparison with KG.

Station	IDW*			OIDW		KG
	$\beta$	MSE	Time (s)	$\beta$ (std.dev.)	SE	Time (s) (std.dev.)
Agigea		26.8233			26.8233	23.92
Albesti		39.1014			39.1016	21.38
Altan Tepe		26.3154			26.3154	23.51
Amzacea		25.5319			25.5319	23.47
Baia		24.9642			24.9642	32.85
Baltagesti		31.246			31.2461	24.62
Biruinta		40.4631			40.4633	26.39
Casian		28.3496			28.3496	18.97
Casimcea		27.3955			27.3956	32.10
Ceamurlia		25.3455			25.3455	19.54
Cerna		28.2859			28.2859	39.45
Cheia		19.4690			19.4690	46.48
Cobadin		30.3138			30.3138	38.22
Corbu		25.2591			25.2591	21.94
Crucea		30.8561			30.8562	26.21
Cuza Voda		23.8778			23.8777	25.27
Daieni		47.0681			47.0681	24.30
Dobromir		29.2771			29.2771	22.91
Dorobantu		35.8369			35.8369	34.72
Greci		26.8270		1.59207	26.8271	4.846
Hamcearca	1.5921	29.0369	190	( $1.1813 \times 10^{-4}$ )	29.0370	(1.0498)
Independenta		35.639			35.6389	18.11
Lipnita		27.4559			27.4559	33.12
Lumina		31.1541			31.1540	36.45
Mihai Viteazu		30.7684			30.7684	74.91
Negru Voda		40.186			40.1860	43.82
Negureni		73.7331			73.7331	2.69
Niculitel		45.067			45.067	48.99
Nuntasi		21.0675			21.0675	29.32
Pantelimon		49.9444			49.9444	21.09
Peceneaga		22.1274			22.1275	38.29
Pecineaga		31.8819			31.8821	33.92
Pestera		35.9176			35.9176	26.19
Pietreni		31.7759			31.7759	22.79
Posta		24.7646			24.7646	21.12
Sacele		20.8606			20.8606	27.64
Saraiu		25.9453			25.9453	23.21
Satu Nou		23.8919			23.8919	47.47
Silistea		22.8196			22.8195	33.66
Topolog		30.4321			30.4321	24.51
Zebil		26.6736			26.6736	29.31

In both scenarios (A1 and B1), OIDW found the optimum, as it was exhaustively identified by IDW. Moreover, the differences between the mean squared errors of the approximations computed with IDW using the  $\beta$  values identified with grid search (IDW\*) and OIDW were not statistically significant. Additionally, the MSEs associated to the IDW algorithms are almost identical; in some cases, these are better than those obtained by KG. The standard deviations of  $\beta$  are very small in both scenarios. This signifies that the PSO search for  $\beta$  is stable—PSO converged in all the 50 runs to very similar values. The average MSE are 30.9327 for IDW\* and OIDW in scenario A1; 30.3704 for IDW\* and 30.3718 for OIDW in scenario B1 and 31.022 for KG, respectively (Tables 1 and 2). It was expected that OIDW in the scenario A1 (Table 1) would yield larger prediction errors than the individual OIDW’s in the scenario B1 (Table 2), since in B1 the algorithm tries to fit a much smaller number of data points.

Nevertheless, the differences with respect to KG are significant only in one case (e.g., Adamclisi). OIDW is run for each station in scenario B1, hence the computational times in Table 2 add up, for a fair comparison; OIDW is run only once for the entire set in A1, with the time as reported in Table 1.

Therefore the time to run the experiments in scenario B1 is smaller than in A1 (1.31 times in for IDW\* and 1.66 times for OIDW).

Tables 3–6 contain the results obtained in experiments performed using the 51 meteorological stations, in scenario A2 and B2, respectively. Results in Table 3 prove that PSO converges to the optimal  $\beta$ , as it was identified by grid search (standard deviation of  $1.1813 \times 10^{-4}$ ).

For the main stations, the standard deviations are equal up to the third decimal for almost all the series. The computational effort, reported as run time, was significantly smaller (40 times) in the case of the OIDW algorithm (190 s for IDW\* and 4.846 s for OIDW). The average MSE computed in A2 experiments were the following: 30.64324 (IDW\*), 30.64315 (OIDW) and 30.876 (KG)—computed using the MSE for the main series; 31.0671 (IDW\*), 31.0671(OIDW) and 29.5480 (KG)—computed using the MSE for the 41 secondary series and 30.9840 (IDW\*), 30.9840 (OIDW) and 29.7924 (KG)—computed using all series. We found that the average MSE was smaller in all situations in scenario A2 for the main series, by comparison to scenario A1. The smallest average MSE in scenarios A1 and A2 corresponded to KG, when extracting only the secondary series (followed by the case when using all the series) for the MSE computation.

The results from Table 5 reveal little to no difference between the  $\beta$ 's identified by grid search and those identified by PSO search. Therefore, the MSE's associated to the computed IDW approximations were identical. The significant difference comes from the computational time: PSO identified the optimal value 50 times faster than the grid search.

The MSE's in scenario B2 are smaller in the experiments with 51 stations than those in the experiments with only ten main stations.

In scenario B1, when taking into account only the results for the main stations, they were as follows: 30.3704 vs. 29.22245 for IDW\*, 30.37178 vs. 29.22245 for OIDW and 31.022 vs. 30.876 (Table 7, rows 3 and 6). This was to be expected because the algorithms received far more information in the scenario B2 than in B1. On the other hand, from the same reason,  $\beta$  st.dev in OIDW increased, while remaining very small.

All the results from Table 7 that summarizes the average MSE in all experiments show similar performances of IDW\*, OIDW and KG. None of the algorithms were the best in all situations, confirming the literature findings [39].

Therefore, one can remark that OIDW yields competitive results in terms of MSE's of estimated values when compared to KG. This is important because it proves OIDW to be a reliable method.

Finding the minimum MSE in the grid search is not straightforward because the MSE evolution patterns are different, as it is illustrated in Figure 5, for two series, where the MSE is computed using all the series, with a grid search step of  $10^{-4}$ . For Cernavoda, the trend is almost linear, while for Adamclisi, it decreases then it increases and presents an inflexion point. Figure 6 depicts the dependence of the average MSE for all series as a function of  $\beta$ . The chart has a similar behavior as for the Adamclisi series.

**Table 5.** Parameter  $\beta$  values obtained in experiments performed using the series recorded at all the meteorological stations in the scenario B2. Comparison with KG.

Station	IDW*			OIDW			KG
	$\beta$	MSE	Time	$\beta$	MSE	Time (s)	MSE
Adamclisi	1.8504	28.9823	4.2	1.8504	28.9823	0.100	31.78
Cernavoda	1.0743	22.0317	4.2	1.07435	22.0317	0.100	25.78
Constanta	1.0001	28.0778	4.1	1.0001	28.0778	0.056	36.47
Corugea	1.4043	17.7585	4.1	1.40429	17.7585	0.110	29.95
Harsova	1.0001	34.104	4.2	1.0001	34.104	0.056	31.51
Jurilovca	1.0001	29.6876	4.3	1.0001	29.6876	0.056	37.43
Mangalia	1.0001	38.6895	4.3	1.0001	38.6895	0.064	27.38
Medgidia	1.0001	23.4367	4.2	1.0001	23.4367	0.058	31.47
Sulina	2.5714	46.4455	4.2	2.57303	46.4455	0.059	28.05
Tulcea	2.9100	23.0109	4.3	2.91002	23.0109	0.071	28.94

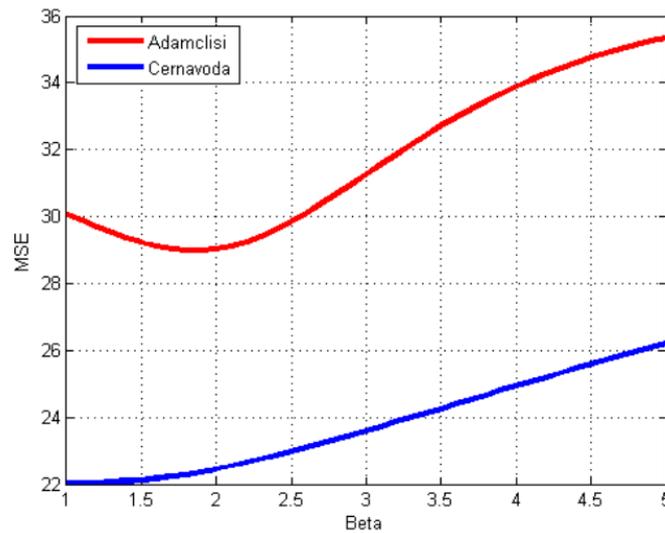
**Table 6.** Parameter  $\beta$  values obtained in experiments performed using the series recorded at all the meteorological stations in the scenario B2. Comparison with KG.

Station	$\beta$	IDW*		B	OIDW		KG
		MSE	Time		MSE	Time (s)	MSE
Agigea	1.8838	26.6137	4.1	1.8838	26.6137	0.070	23.92
Albesti	3.4336	35.5855	4.1	3.43379	35.5855	0.067	21.38
Altan Tepe	1.5072	26.3112	4.1	1.50717	26.3112	0.120	23.51
Amzacea	1.4064	25.4654	4.2	1.40655	25.4654	0.099	23.47
Baia	1.722	24.9073	4.3	1.72191	24.9073	0.120	32.85
Baltagesti	2.3856	30.1228	4.3	2.38567	30.1228	0.086	24.62
Biruinta	3.1188	36.8629	4.3	3.11882	36.8629	0.110	26.39
Casian	1.6395	28.346	4.3	1.6394	28.346	0.100	18.97
Casimcea	2.3905	26.6605	4.2	2.39077	26.6605	0.084	32.1
Ceamurlia	1.5454	25.3378	4.3	1.54547	25.3378	0.086	19.54
Cerna	1.8537	28.126	4.2	1.85357	28.126	0.084	39.45
Cheia	1.6433	19.4619	4.1	1.64323	19.4619	0.120	46.48
Cobadin	1.0001	30.1472	4.1	1.0001	30.1472	0.055	38.22
Corbu	1.5919	25.2591	4.2	1.59192	25.2591	0.110	21.94
Crucea	2.3662	29.745	4.1	2.36621	29.745	0.120	26.21
Cuza Voda	1.0001	23.4537	4.1	1.0001	23.4537	0.067	25.27
Daieni	5	44.9327	4.1	5	44.9327	0.065	24.3
Dobromir	1.4430	29.2665	4.2	1.44297	29.2665	0.072	22.91
Dorobantu	1.3107	35.8154	4.2	1.31057	35.8154	0.097	34.72
Greci	3.8756	25.9016	4.2	3.87535	25.9016	0.087	25.23
Hamcearca	3.6478	26.9690	4.2	3.648	26.9690	0.077	22.56
Independenta	1.0001	33.7906	4.2	1.0001	33.7906	0.057	18.11
Lipnita	1.6913	27.4506	4.5	1.6900	27.4506	0.08	33.12
Lumina	1.0001	30.1164	4.4	1.0001	30.1164	0.066	36.45
Mihai Viteazu	1.4875	30.7659	4.2	1.48901	30.7659	0.071	74.91
Negru Voda	1.0001	39.9995	4.3	1.0001	39.9995	0.061	43.82
Negureni	2.3998	73.5851	4.2	2.39992	73.5851	0.093	2.69
Niculitel	1.1157	44.7084	4.2	1.11566	44.7084	0.11	48.99
Nuntasi	1.6074	21.067	4.2	1.60745	21.067	0.11	29.32
Pantelimon	5	45.7025	4.5	5	45.7025	0.061	21.09
Peceneaga	3.4614	19.9815	4.4	3.46124	19.9815	0.081	38.29
Pecineaga	3.1474	29.0975	4.4	3.14738	29.0975	0.092	33.92
Pestera	1.0001	35.5876	4.4	1.0001	35.5876	0.057	26.19
Pietreni	1.5219	31.772	4.5	1.52222	31.772	0.082	22.79
Posta	1.7224	24.747	4.2	1.7224	24.747	0.100	21.12
Sacele	1.5402	20.8558	4.2	1.54022	20.8558	0.089	27.64
Saraiu	2.9462	25.1995	4.2	2.94627	25.1995	0.073	23.21
Satu Nou	1.1494	23.694	4.1	1.14939	23.694	0.092	47.47
Silistea	1.1135	22.4699	4.2	1.11361	22.4699	0.084	33.66
Topolog	2.6361	29.9864	4.3	2.63624	29.9864	0.072	24.51
Zebil	1.6260	26.6716	4.2	1.6260	26.6716	0.092	29.31

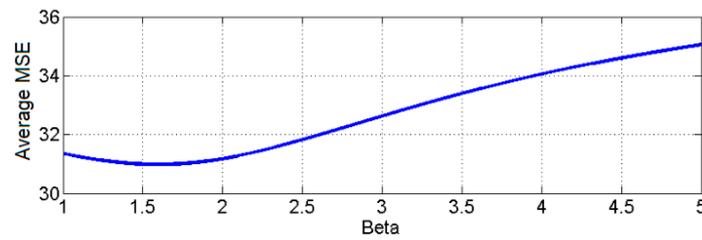
**Table 7.** Mean values of standard error (average mean squared error (MSE)) over all the series in different scenarios.

Scenario	IDW*		OIDW		KG <sup>1,2</sup>
	A1 <sup>1</sup>	B1 <sup>2</sup>	A1 <sup>1</sup>	B1 <sup>2</sup>	
Main stations	30.93269	30.3704	30.93269	30.37178	31.022
Scenario	IDW*		OIDW		KG
	A2	B2	A2	B2	
Main stations	30.64324 <sup>3</sup>	29.22245 <sup>5</sup>	30.64315 <sup>3</sup>	29.22245 <sup>5</sup>	30.876 <sup>3</sup>
Secondary stations	31.0671 <sup>4</sup>	30.30585 <sup>6</sup>	31.0671 <sup>4</sup>	30.30585 <sup>6</sup>	29.5280 <sup>4</sup>
All stations	30.9840 <sup>3,4</sup>	30.0934 <sup>5,6</sup>	30.9840 <sup>3,4</sup>	30.0934 <sup>5,6</sup>	29.7924 <sup>3,4</sup>

Note: <sup>1-6</sup> means that the values have been computed using the values from Tables 1–6, respectively. <sup>3,4</sup> means that the values have been computed using the values from Tables 3 and 4. <sup>5,6</sup> means that the values have been computed using the values from Tables 5 and 6



**Figure 5.** MSE’s for Adamclisi and Cernavoda series in a grid search with a step of  $10^{-4}$  on the study interval.



**Figure 6.** Average MSE computed as the average of the MSEs corresponding to all the series in a grid search with a step of  $10^{-4}$  on the study interval.

With respect to computational effort: for detecting the  $\beta$  that minimizes the MSE for fitting the values of a series in IDW\* using a grid search with a step  $r$ , on an interval  $[a, b]$ , the computation should be performed  $[(b - a) / r] + 1$  times (where  $[]$  is the integer part of the number inside the bracket). For example, for a grid search with a step of 0.1, IDW\* was performed 50 times for each station, so 500 times when only the main stations were employed and 2550 times when all the 51 series were used.

Once again, taking into account the consumed time in scenarios A1 or A2, B1 or B2 is more convenient to run OIDW. This idea is supported by the fact that the  $\beta$ 's identified with OIDW in all the runs were identical to optimal values of  $\beta$  found by grid search IDW\* up to the third digits (so, OIDW converged to a global optimum, in all cases).

OIDW performed better than KG in 60% of cases in scenario A1, in 80% of cases in scenario B1, and in 49.01% cases in scenarios A2 and B2. While in the scenarios A1 and B1, the OIDW MSE and KG MSE are comparable, in the A2 and B2 scenarios, there are some discrepancies between them, as for example, for Sulina, Cheia, Negureni, Pantelimon and Mihai Viteazu (in A2 and B2). This situation could be explained by the following reasons: (a) the low number of series used in A1 and B1 scenario, (b) the inhomogeneity of the stations on the in the Dobrogea region, (c) the climate differences on the different part of the region and (d) the possible anisotropy was not considered.

The results of the statistical tests on the SE series for OIDW and IDW\* are presented in the following. The normality test for the MSE series corresponding to the experiments in scenarios A1, B1, OIDW, KG and IDW\* from Tables 1 and 2, A2, B2, IDW\*, KG from Tables 3 and 5, KG from Table 3 + Table 4 and KG from Table 5 + Table 6 could not reject the normality hypothesis. The same test led to the normality rejection for the IDW\* MSE and OIDW MSE series corresponding to the secondary stations from Tables 4 and 6, but not for OIDW MSE corresponding to all the series in Table 3 + Tables 4 and 5 + Table 6.

A spatial map containing illustration the distribution of MSE in OIDW form B2 scenario (values from Table 5 + Table 6) is presented in Figure 7. The highest MSEs correspond to the station near the region border, where the stations had few close neighbors.

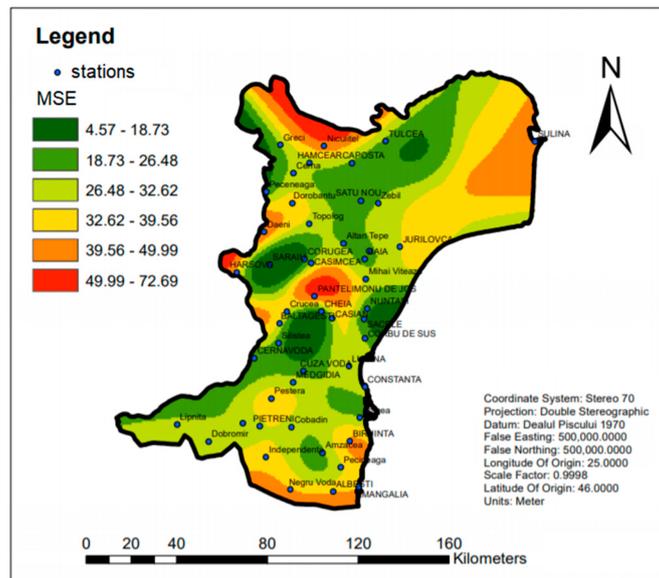


Figure 7. MSE in OIDW (scenario B2).

To offer a different perspective of the OIDW performance, we computed the Mean Absolute Percentage Error (MAPE) and the Kling–Gupta efficiency (KGE).

The MAPE is given by the formula:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{p_i - m_i}{m_i} \right| \tag{16}$$

where  $n$  is number of stations used in the IDW,  $p_i$  is the value predicted by IDW for the precipitation at station  $i$  and  $m_i$  is the measured value for the precipitation at station  $i$ .

The KGE coefficient is introduced in [40]; it is defined by formula:

$$KGE = 1 - ED \tag{17}$$

where:

$$ED = \sqrt{(r - 1)^2 + \left(\frac{\sigma_s}{\sigma_0} - 1\right)^2 + \left(\frac{\mu_s}{\mu_0} - 1\right)^2} \tag{18}$$

where  $\mu_s$  is mean of the values resulted from the model,  $\mu_0$  is the mean of the recorded values,  $\sigma_s$  is the standard deviation of the values resulted from the model and  $\sigma_0$  standard deviation of the recorded values.

MAPE is a scale-independent indicator that could be used in order to compare the performance of a given method on separate data sets. The lower MAPE is, the better the model is. The KGE coefficient is dimensionless as well and has an ideal value of unity. Although, in this study, the models were calibrated using the MSE as the model calibration criterion, the KGE and MAPE provide further validation of the obtained models.

Table 8 provides the MAPE values together with the Root Mean Square Errors ( $RMSE = \sqrt{MSE}$ ) and the KGE for the ten main stations.

**Table 8.** Mean Absolute Percentage Error (MAPE) values for the predictions computed with OIDW for the series recorded at all the meteorological stations in the scenarios A1 and B1.

Station	Scenario A1			Scenario B1		
	RMSE	MAPE	KGE	RMSE	MAPE	KGE
Adamclisi	5.6937	0.2197	0.5285	5.3458	0.2104	0.6339
Cernavoda	4.7874	0.1788	0.6711	4.7626	0.1736	0.6696
Constanta	5.5068	0.2299	0.5370	5.4928	0.2254	0.5353
Corugea	4.7335	0.1753	0.6847	4.7253	0.1745	0.6799
Harsova	5.9689	0.3490	0.4525	5.9377	0.3372	0.5139
Jurilovca	4.9094	0.2760	0.6529	4.8971	0.2754	0.6549
Mangalia	6.4840	0.2601	0.3447	6.4770	0.2588	0.3418
Medgidia	4.7308	0.2026	0.7105	4.7231	0.2019	0.7202
Sulina	6.6348	0.7615	0.2210	6.5888	0.7566	0.2400
Tulcea	5.7519	0.2454	0.4451	5.7489	0.2443	0.4457

Similar results for MAPE and KGE were obtained in the A2 and B2 scenarios, so we are not presenting them here.

The highest MAPE value (so the worst data fit) was noticed for the Sulina station in both scenarios. It was expected, since Sulina is situated 12 km offshore, in the Danube Delta, and the climate presents different particularities compared with the rest of Dobrogea region. The same was true for the KGE coefficient: its lowest values were obtained for the Sulina and Mangalia stations (the most isolated stations).

Since the values of both goodness-of-fit indicators, dimensional-RMSE and dimensionless-MAPE are small, and most of the KGE values are bigger than 0.5, it resulted in the OIDW performing well.

The Levene’s test could not reject the hypotheses that groups of MSE from Tables 1–4 (together), Tables 5 and 6 (together) and Tables 3–6 were homoskedastic, the corresponding *p*-values associated to this test were greater than the significance level of 5%.

The Kruskal–Wallis test could not reject the null hypothesis, the corresponding *p*-values being 0.9356 (Tables 1 and 2), 0.8348 (Table 3 + Table 5), 0.6437 (Table 4 + Table 6) and 0.7519 (Tables 3–6 all the stations), respectively. The eqdist.etest test applied to the same groups of series as in the previous test could not reject the hypothesis that each group of series has the same distribution, the corresponding *p*-values being greater than the significance level of 5%.

Concluding, the MSE’s of the estimations obtained by OIDW, IDW\* and KG were not significantly different.

When KG is not very straightforward to be applied and searching for the best parameter  $\beta$  for IDW with grid search is computationally intensive, OIDW proves to be a convenient and rapid to use solution.

Other optimization methods can be applied for the power parameter of IDW calibration, such as methods based on the Golden Section (GS) method, provided that the hypotheses of this method are satisfied. As an example, we tried to calibrate the parameter  $\beta$  by this additional optimization method, namely by the fminbnd Matlab function [41]. This function applies GS, followed by parabolic interpolation; the boundaries for the search are to be specified as parameters. In the experiments performed on the series from the ten main stations with the fminbnd method, we obtained an approximation of 1.1903 *f* or  $\beta$  with a duration of 1 s, with MSE = 30.75. The results are comparable in this situation with those obtained by both OIDW and IDW. The assumptions for this method include that the function to be minimized should be continuous, which is hardly the case in real world problems. Additionally, since in some situations the fminbnd fails to find the optimum (as in the case of independent stations, where there are local minima or when minimum lies on the boundary of the search domain [41]), we consider that an extensive study should be done to compare our approach with that of fminbnd. The results will be communicated in another article.

## 5. Conclusions

The paper describes a hybrid PSO-IDW algorithm for the modeling of data; it is applied on maximum precipitation data in the experimental part. The algorithm is based on a well-known spatial interpolation method (IDW). Since the application of IDW makes use of the power parameter  $\beta$ , we proposed the use of the PSO algorithm to identify a suitable value for this parameter. Empirical results proved that the IDW performance was maintained, while the new method offered lower computational costs. Results of the IDW and OIDW were further compared to Kriging.

The optimization of IDW with PSO described in this paper offered an alternative to an exhaustive search for the problem of tuning the power parameter in the IDW method. In cases when the estimations provided by the traditional application of IDW are good, the results obtained with OIDW should preserve the quality of the prediction and be better in terms of computational effort.

Kriging is a powerful geostatistical method, suitable to be applied in certain cases, and its application assumes profound knowledge of spatial statistics. When the spatial correlation is strong and the variogram can depict well the spatial variability, the KG algorithm predicts very well. Even so, for our precipitation data, the results are not substantially better by KG, as it was the case for other environmental variables studied in [17]. In cases where the nature of the input data does not support KG, instead of using IDW for prediction it is wiser to choose OIDW since it automates the process of identifying the appropriate power parameter, with smaller computational effort. Our approach also benefits from the advantages of the PSO algorithm: a simple implementation, robustness, small number of parameters to adjust and a high probability and efficiency in finding the global optima, fast convergence, short computational time and modeling accuracy. OIDW is easy to use by people that do not have statistical knowledge [29,42].

The limitations of IDW are preserved by OIDW: in locations with sparse neighbors (such as those near the borders of the study region), the errors for IDW simulated data are larger. While providing a solution for optimizing the IDW parameter, OIDW uses PSO, which has several parameters of its own that need to be set (e.g., swarm size, the weights in the Equations (7) and (8)). In our experiments, setting these parameters to values commonly used in the literature led to very good results.

The contribution of our paper was two-fold. For hydrologists, we introduced an optimization of a well-known and widely used spatial interpolation method—namely IDW. For the artificial intelligence community, we proved the utility of the application of the PSO metaheuristic on a parameter optimization problem.

The results obtained in our case study using time series data from 51 meteorological locations in the Dobrogea region (Romania) encouraged us to consider OIDW as a good choice for the spatial interpolation of precipitation data. Still, more experiments will be performed, using various types of environmental data and various measures of performance of the models, in order to decide the appropriateness of OIDW as a general spatial interpolator.

**Author Contributions:** Data curation: A.B. (Alina Barbulescu); Investigation: A.B. (Alina Barbulescu) and A.B. (Andrei Bautu); Methodology; E.B.; Software: A.B. (Andrei Bautu); Supervision: A.B. (Alina Barbulescu); Writing original draft: A.B. (Alina Barbulescu) and E.B.; Writing—review and editing: A.B. (Alina Barbulescu) and E.B. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Acknowledgments:** We thank the anonymous reviewers for the very useful comments and insights offered in the reviews. Our paper was significantly improved by taking into account their observations.

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

## References

1. Ly, S.; Charles, C.; Degre, A. Different methods for spatial interpolation of rainfall data for operational hydrology and hydrological modeling at watershed scale: A review. *Biotechnol. Agron. Soc. Environ.* **2013**, *17*, 67–82.
2. Li, J.; Heap, A.D. Spatial interpolation methods applied in the environmental sciences: A review. *Environ. Model. Softw.* **2014**, *53*, 173–189. [[CrossRef](#)]
3. Li, J.; Heap, A.D.; Potter, A.; Daniell, J.J. Application of machine learning methods to spatial interpolation of environmental variables. *Environ. Model. Softw.* **2011**, *26*, 1647–1659. [[CrossRef](#)]
4. Lu, G.Y.; Wong, D.W. An adaptive inverse-distance weighting spatial interpolation technique. *Comput. Geosci.* **2008**, *34*, 1044–1055. [[CrossRef](#)]
5. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [[CrossRef](#)]
6. Chang, C.L.; Lo, S.L.; Yu, S.L. The parameter optimization in the inverse distance method by genetic algorithm for estimating precipitation. *Environ. Monit. Assess.* **2006**, *117*, 145–155. [[CrossRef](#)]
7. Kumari, M.; Ashoke, B.; Oinam, B.; Singh, C.K. Comparison of Spatial Interpolation Methods for Mapping Rainfall in Indian Himalayas of Uttarakhand Region. In *Geostatistical and Geospatial Approaches for the Characterization of Natural Resources in the Environment*; Janardhana Raju, N., Ed.; Springer: Cham, Switzerland, 2016; pp. 159–168.
8. Chen, T.; Ren, L.; Yuan, F.; Yang, X.; Jiang, S.; Tang, T.; Liu, Y.; Zhao, C.; Zhang, L. Comparison of Spatial Interpolation Schemes for Rainfall Data and Application in Hydrological Modeling. *Water* **2017**, *9*, 342. [[CrossRef](#)]
9. Cavus, Y.; Aksoy, H. Spatial Drought Characterization for Seyhan River Basin in the Mediterranean Region of Turkey. *Water* **2019**, *11*, 1331. [[CrossRef](#)]
10. Noori, M.J.; Hassan, H.; Mustafa, Y.T. Spatial estimation of rainfall distribution and its classification in Duhok governorate using GIS. *J. Water Res. Protect* **2014**, *6*, 75–82. [[CrossRef](#)]
11. Golkhatmi, N.S.; Sanaeinejad, S.H.; Ghahraman, B.; Pazhand, H.R. Extended modified inverse distance method for interpolation rainfall. *Int. J. Eng. Invent.* **2012**, *1*, 57–65.
12. Barbulescu, A. A new method for estimation the regional precipitation. *Water Resour. Manag.* **2016**, *30*, 33–42. [[CrossRef](#)]
13. Gholipour, Y.; Shahbazi, M.M.; Behnia, A.R.A.S.H. An improved version of Inverse Distance Weighting metamodel assisted Harmony Search algorithm for truss design optimization. *Latin Am. J. Solids Struct.* **2013**, *10*, 283–300. [[CrossRef](#)]
14. Zhang, X.; Liu, G.; Wang, H.; Li, X. Application of a hybrid interpolation method based on support vector machine in the precipitation spatial interpolation of basins. *Water* **2017**, *9*, 760. [[CrossRef](#)]
15. Nourani, V.; Behfar, N.; Uzelaltinbulat, S.; Sadikoglu, F. Spatiotemporal precipitation modeling by artificial intelligence-based ensemble approach. *Environ. Earth Sci.* **2020**, *79*, 6. [[CrossRef](#)]
16. Bărbulescu, A. Modeling temperature evolution. Case study. *Rom. Rep. Phys.* **2016**, *68*, 788–798.
17. Bărbulescu, A. Models for temperature evolution in Constanta area (Romania). *Rom. J. Phys.* **2016**, *3–4*, 676–686.
18. Shepard, D. A two-dimensional interpolation function for irregularly-spaced data. In Proceedings of the 23rd ACM national conference, Las Vegas, NV, USA, 27–29 August 1968; pp. 517–524.
19. Chiles, J.-P.; Delfiner, P. *Geostatistics. Modeling Spatial Uncertainty*, 2nd ed.; Wiley: Hoboken, NJ, USA, 2012.
20. Cressie, N. *Statistics for Spatial Data*; John Wiley & Sons, Inc.: Chichester, UK, 1993.
21. Hartmann, K.; Krois, J.; Waske, B. E-Learning Project SOGA: Statistics and Geospatial Data Analysis. Department of Earth Sciences, Freie Universitaet Berlin, 2018. Available online: <https://www.geo.fu-berlin.de/en/v/soga/Overview-and-Structure/index.html> (accessed on 10 January 2020).
22. Bailey, T.C.; Gatrell, A.C. *Interactive Spatial Data Analysis*; Longman Scientific & Technical: Essex, UK, 1995.
23. Yasrebi, J.; Saffari, M.; Fathi, H.; Karimian, N.; Moazallahi, M.; Gazni, R. Evaluation and comparison of ordinary kriging and inverse distance weighting methods for prediction of spatial variability of some soil chemical parameters. *Res. J. Biol. Sci.* **2009**, *4*, 93–102.
24. Nusreta, D.; Dugb, S. Applying the inverse distance weighting and kriging methods of the spatial interpolation on the mapping the annual precipitation in Bosnia and Herzegovina. In Proceedings of the International Congress on Environmental Modelling and Software, Sixth Biennial Meeting, Leipzig, Germany, 11 January 2012; Seppelt, R., Voinov, A.A., Lange, S., Bankamp, D., Eds.; 2012.

25. Forrester, A.I.J.; Söbester, A.; Keane, A.J. *Engineering Design via Surrogate Modelling: A Practical Guide*; John Wiley & Sons, Ltd.: Chichester, UK, 2008.
26. Luke, S. *Essentials of Metaheuristics*, 2nd ed.; Lulu: Raleigh, NC, USA, 2013; Available online: <https://cs.gmu.edu/~sean/book/metaheuristics/> (accessed on 1 October 2019).
27. Blum, C.; Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Comput. Surv.* **2003**, *35*, 268–308. [[CrossRef](#)]
28. Kennedy, J. Particle Swarm Optimization. In *Encyclopedia of Machine Learning*; Sammut, C., Webb, G.I., Eds.; Springer: New York, NY, USA, 2011; pp. 760–766.
29. Hasanien, H.M. Particle swarm design optimization of transverse flux linear motor for weight reduction and improvement of thrust force. *IEEE Trans. Ind. Electron.* **2011**, *58*, 4048–4056. [[CrossRef](#)]
30. Trelea, I.C. The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Inf. Process. Lett.* **2003**, *85*, 317–325. [[CrossRef](#)]
31. Adhikari, R.; Agrawal, R.; Kant, L. PSO based Neural Networks vs. Traditional Statistical Models for Seasonal Time Series Forecasting. In Proceedings of the 3rd IEEE International Conference (IACC), Ghaziabad, India, 22–23 February 2013; IEEE; pp. 719–725. Available online: <https://www.worldcat.org/title/proceedings-of-the-2013-3rd-ieee-international-advance-computing-conference-iacc-february-22-23-2013-ghaziabad-india/oclc/875926765> (accessed on 10 October 2019).
32. Nickabadi, A.; Ebadzadeh, M.M.; Safabakhsh, R. A novel particle swarm optimization algorithm with adaptive inertia weight. *Appl. Soft Comput.* **2011**, *11*, 3658–3670. [[CrossRef](#)]
33. Birge, B. Particle Swarm Optimization Toolbox. MATLAB Central File Exchange. Available online: <https://www.mathworks.com/matlabcentral/fileexchange/7506-particle-swarm-optimization-toolbox> (accessed on 1 January 2020).
34. Paul, H. Package ‘Automap’. Available online: <https://cran.r-project.org/web/packages/automap/automap.pdf> (accessed on 15 January 2020).
35. Pebesma, E.J. Gstat User’s Manual. Available online: <http://www.gstat.org/gstat.pdf> (accessed on 15 November 2019).
36. Barbulescu, A. *Studies on Time Series. Applications in Environmental Sciences*; Springer: Basel, Switzerland, 2016.
37. Kruskal, W.H.; Wallis, W.A. Use of Ranks in One-Criterion Variance Analysis. *J. Am. Stat. Assoc.* **1952**, *47*, 583–621. [[CrossRef](#)]
38. Szekely, G.J.; Rizzo, M.L. Testing for Equal Distributions in High Dimension. Available online: <http://interstat.statjournals.net/YEAR/2004/articles/0411005.pdf> (accessed on 19 November 2018).
39. Gong, G.; Mattevada, S.; O’Byrant, S. Comparison of the accuracy of kriging and IDW interpolations in estimating groundwater arsenic concentrations in Texas. *Environ. Res.* **2014**, *130*, 59–69. [[CrossRef](#)]
40. Gupta, H.V.; Kling, H.; Yilmaz, K.K.; Martinez, G.F. Decomposition of the mean squared error and NSE performance criteria: Implications for improving hydrological modeling. *J. Hydrol.* **2009**, *377*, 80–91. [[CrossRef](#)]
41. Fminbnd Method for Optimization. Available online: <https://www.mathworks.com/help/matlab/ref/fminbnd.html> (accessed on 10 March 2020).
42. Abdmouleh, Z.; Gastli, A.; Ben-Brahim, L.; Haouari, M.; Al-Emadi, N.A. Review of optimization techniques applied for the integration of distributed generation from renewable energy sources. *Renew. Energy* **2017**, *117*, 266–280. [[CrossRef](#)]

