Large-scale rain gauge network optimization using a kriging emulator

Henriksen, Rasmus Lau Thejlade; Hubrechts, Jonas Bruun; Møller, Jan Kloppenborg; Knudsen, Per; Pedersen, Jonas Wied

Published in:
Journal of Hydrology

Link to article, DOI:
10.1016/j.jhydrol.2024.131360

Publication date:
2024

Document Version
Peer reviewed version

Citation (APA):

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.
Highlights

1. A novel framework for handling large-scale rain gauge network optimization problems
2. An emulator replaces Ordinary Kriging within a fast optimization framework
3. It is used on Denmark’s national network and places 175 new gauges in 15 minutes

Abstract

Rain gauge networks deliver crucial observations for many water-related applications but can be expensive to purchase and operate, which makes optimization of the number and locations of gauges an important task. Traditional optimization approaches often focus on kriging-based methods that are computationally expensive, which limits the scale of the optimization to small areas or a very limited number of gauges. This study presents a novel workflow with high computational efficiency that is able to handle optimization problems on large rain gauge networks. This is accomplished by developing a fast parametric emulator of the commonly used ordinary kriging approach. The results show that the developed emulator is able to accurately reproduce the original kriging uncertainty estimates with a computational speed up of a factor of 3000. In order to determine the best locations for new gauges, a greedy optimization heuristic that relies on sequential placement of gauges is developed. The sequential optimization can lead to sub-optimal solutions by itself, so a resubstitution mechanism is introduced to correct for this. The workflow is applied to the national gauging network of Denmark.
with 291 gauges, where it is able to optimally place 175 new gauges within 1 hour of running time. A similar optimization with a traditional kriging approach would have taken approximately 125 days to complete highlighting the value the workflow.
Large-scale rain gauge network optimization using a kriging emulator

Rasmus Lau Thejlade Henriksen\textsuperscript{a,b}, Jonas Bruun Hubrechts\textsuperscript{a,b}, Jan Kloppenborg Møller\textsuperscript{b}, Per Knudsen\textsuperscript{c}, Jonas Wied Pedersen\textsuperscript{a,d}

\textsuperscript{a}Weather research, Danish meteorological institute, Lyngbyvej 100, Copenhagen, 2100, Denmark
\textsuperscript{b}Department of applied computer science and mathematics (DTU Compute), Technical university of Denmark, Richard Petersens Plads, Building 324, Kgs. Lyngby, 2800, Denmark
\textsuperscript{c}Department of Space Research and Technology (DTU Space), Technical University of Denmark, Elektrovej, Building 327, Kgs. Lyngby, 2800, Denmark
\textsuperscript{d}Department of environmental and resource engineering (DTU Sustain), Technical University of Denmark, Bygningstorvet, Building 115, Kgs. Lyngby, 2800, Denmark

Keywords: Rain gauge network design, Ordinary Kriging, Geostatistics, Emulator, Large-scale optimization

1. Introduction

High-quality rainfall observations are crucial to a large range of disciplines, such as water resources management, agriculture, climatology and early flood warning systems. These observations are typically obtained from a network of rain gauges on the ground, which provides time series of rainfall at specific geographical locations. Remote sensing technology, such as weather radars and satellites, can provide spatially distributed rainfall data, but typically have to be validated against and/or merged with in-situ rain gauge observations (Cecinati et al., 2018; Ochoa-Rodriguez et al., 2019; Wu et al., 2020). Rainfall is a process that exhibits high temporal and spatial variability (Peleg et al., 2018), and to adequately capture these phenom-
ena rain gauge networks have to be of sufficient size and density for a given
use case (Merino et al., 2021; Nikolopoulos et al., 2015). A large network
of gauges is, however, expensive to purchase and maintain. This leads to
the problem of designing good configurations of rain gauge networks with
an optimal number and location of gauges within a given area. In general,
there are three different categories of rain gauge network design problems
(Chacon-Hurtado et al., 2017):

- Reduction: Finding an optimal subset of gauges in an existing network
  in order to save costs by removing the most redundant gauges (e.g.
  Bayat et al. (2021a); Ghomlaghi et al. (2023)).

- Augmentation: Finding new optimal locations for additional gauges
to maximize the benefit of an investment into better rainfall data sets
  (e.g. Chen et al. (2008); Xu et al. (2018)).

- Relocation: Relocating the most redundant gauges to more optimal
  locations in the network (e.g. Adhikary et al. (2015)).

See Simoyama et al. (2022) for an exhaustive review of studies in each of the
three categories.

A reduction exercise can be done purely based on time series data col-
clected by the existing rain gauge network. Augmentation or relocation, how-
ever, requires the ability to predict how potential, new gauges will impact
the performance of the overall network. Therefore approaches to optimal
rain gauge network augmentation involves setting up a framework with the
following main steps (adapted from Simoyama et al. (2022)):
1. Developing a statistical model that allows for spatial interpolation at locations between existing gauges, often including estimates of the uncertainty on the interpolated values.

2. Defining candidate locations for placing new gauges or relocating existing gauges to.

3. Defining one or multiple cost functions that describes the performance of the network and thus encapsulates what a “good” network means to the user.

4. Developing an optimization algorithm that use the statistical model to explore new network configurations with the aim of optimizing the cost function.

In step 1, the purpose of the statistical model is to assess how changes to the network configuration (adding, relocating or removing gauges) impacts e.g. the accuracy or precision of the interpolated rainfall values. A common choice for the statistical model is often a variant of the geostatistical technique ”kriging”, which have been shown to outperform deterministic methods such as Inverse Distance Weighting for spatial interpolation of rainfall (Adhikary et al., 2017). Another advantage of kriging over deterministic methods is that it provides information about the precision of the interpolated rainfall estimates through the kriging variance. A common example of a kriging variant being used for rain gauge network assessments is ordinary kriging (Chen et al., 2008; Pardo-Igúzquiza, 1998; Xu et al., 2018), while another study also has investigated ordinary co-kriging and kriging with external drift (KED) (Ghomlaghi et al., 2023). Ordinary kriging for rain gauge network assessment have also been implemented within frameworks both for
point estimates and spatial averages, which is also known as block kriging (Bayat et al., 2021b). The statistical model can be applied to observations of different temporal aggregations, such as annual precipitations sums (e.g. Attar et al. (2019); Ghomlaghi et al. (2023)) or daily sums (Xu et al., 2018; Wang et al., 2020). Most studies view the observed data as "ground truth" but some have developed frameworks that can treat the observations as uncertain data points (Bayat et al., 2021a).

Step 2 is motivated by the fact that placing new gauges in an augmentation or relocation exercise has to confront the problem that there is an infinite number of candidate locations for new gauges if geographical x- and y-dimensions are treated as continuous variables. It is therefore very common to constrain the problem by only considering a discrete set of new candidate locations. In the literature, this is often done by experts manually choosing a set of locations that a priori seem like good options. Due to their manual nature, these approaches tend to only deal with relatively small augmentations (typically 1-5 gauges) to an existing network (Adhikary et al., 2015; Xu et al., 2018). Another common choice is to specify candidate locations as the centroids of a rectangular grid over a given study area (Wang et al., 2020; Yeh et al., 2011).

The choice of cost function for rain gauge network optimization in step 3 is a whole study in itself (Foroozand and Weijs, 2021), and examples of cost functions from the literature involves improving estimation accuracy, reducing estimation uncertainty (i.e. minimizing prediction error variance), maximizing information content or a combination of metrics within multi-objective optimization frameworks (Volkmann et al., 2011; Wang et al., 2020;
Trying to place several new gauges to augment an existing network leads to a combinatorial optimization problem where the computational complexity considerably increases as the network size and the number of new gauges to be placed increases. The choice of optimization algorithm in step 4 has to solve this issue. The two-stage optimization approach of Bayat et al. (2021b) handles the issue by drastically reducing the number of possible locations for new rain gauges in a first, fast scan into so-called large geographical "blocks". Once the fast, large-scale prioritization of new gauges into each large block has been done, they then perform small-scale optimization of the optimal locations within each block. Other studies use large global optimization frameworks such as simulated annealing (Pardo-Igúzquiza, 1998), NSGA-II (Wang et al., 2020) and artificial bee colony optimization (Attar et al., 2019). These types of techniques require many simulations to explore the full solution space and converge on a global optimum. Due to the computational expense of traditional kriging algorithms, they are therefore often limited to dealing with small gauging networks or are only able to add a few gauges to the existing networks. If a problem requires analyzing large gauging networks (e.g. at the national scale) and testing scenarios with large augmentations (perhaps in the order of hundreds of new gauges) there is very little help in the scientific literature on how to do this.

The objective of this study is therefore to present a workflow with high computational efficiency in order to place many new rain gauges in a classic rain gauge network augmentation task. Specifically, we develop novel contributions to improve the efficiency of the above-mentioned steps 1 and 4. To
do so, we:

- Develop a novel algorithm that can emulate the effect of placing new gauges in the popular, yet computationally intensive ordinary kriging framework.
- Develop a greedy heuristic with a resubstitution mechanism for fast optimization of new network configurations by minimizing the average kriging variance.
- Test the proposed workflow on the large national rain gain network of Denmark.

The paper is structured as follows: Section 2 gives an overview of the available data. Section 3 explains the theory needed to understand the ordinary kriging process, describes the developed framework for fast emulation of ordinary kriging, and introduces a optimization heuristic for fast placements of many new rain gauges. Section 4 presents the results and discussion of applying the presented workflow. The conclusions are presented in Section 5.

2. Data and case study

The case study of this paper is the national rain gauge network of Denmark, which is operated and quality controlled by the Danish Meteorological Institute (DMI). The network originally consisted of rain gauge buckets where contracted, local operators recorded daily sums of precipitation from 1860 until the late 2000s. From the late 2000s until 2011, the gauging equipment was
gradually modernized and upgraded to automatic weighing rain gauges that can provide minute-resolution rainfall intensities. To offset the expenses of installing and operating the modern equipment, the total number of DMI-owned gauging stations in the network was severely cut from around 600 before the change to around 100 now. The new network was augmented with tipping bucket rain gauges from local wastewater utility companies that DMI already had the responsibility of quality assuring (Jørgensen et al., 1998). The total number of gauges in the current network (year 2022) is 291 gauges, but due to the large fraction of utility-owned gauges, the network is now severely spatially non-uniform with most gauges concentrated around major urban areas. Figure 1(a) and (b) shows the locations of gauges in the current and old networks, respectively. In recent years, hydrological applications of interpolated rainfall estimates from the new, non-uniform network have questioned whether the observations in rural areas are of sufficient quality for hydrological water balance closure (Thodsen et al., 2020). DMI is therefore exploring options for an upgrade to the network, which would mean investing in many more rain gauges “to close the largest rural gaps”.

This study employs three different types of data, all courtesy of DMI:

1. **Rain gauge data:** Annual rainfall sums from individual gauges and station metadata, including geographical locations of gauges, start and end dates for their operational periods, and any information from DMI quality assurance (e.g. down time due to maintenance or detected outliers).

2. **DMI’s official gridded precipitation product:** DMI publishes a gridded open access data set of rainfall sums at various temporal resolutions.
(daily, monthly, annual) called "ClimateGridDK" (Scharling and Kern-Hansen, 2012). The open access data is available in a spatial resolution of 10x10km$^2$, but for this study we have access to DMI’s internal 1x1km$^2$ product. The interpolated maps are created using an Inverse Distance Weighting (IDW) algorithm, which weighs nearby gauges with \( \frac{1}{d^x} \), where \( x \) is a tuning parameter that DMI’s climatologists define. Afterwards, the maps are smoothed by a Gaussian kernel (Figure 1(c)).

3. Coastal-inland climate classification: As part of DMI’s inverse-distance algorithm, they account for different climatic conditions across the country and how that affects the correlation of rainfall between gauges. For this purpose, the climatologists at DMI have developed a classification of each 1x1km$^2$ grid cell in the interpolated product on a scale from 0 (coastal) to 100 (inland). In DMI’s IDW algorithm, the difference in climate class value is used as an extra "distance" added to the geographical distance between two gauges (Figure 1(d)).
In this study, the optimization of new rain gauge locations is performed on the current network configuration, and the applied algorithms are therefore trained on data from the period 2016-2021, since there have not been any major changes to the network in that period. The types of changes that can occur include setting up new gauges, removal of existing gauges, or moving gauges to other locations. These changes can occur throughout the year and influence the final annual precipitation sums. Stations that have not been at the same location for at least 360 days in a given year are therefore removed.
from the data set. Gauges that have been identified by DMI’s official quality control as giving faulty data for more than five days in a year are removed as well. On average this filtering removes approximately 20% of the rain gauges each year.

3. Methods

The presented framework for performing large-scale rain gauge network design can be summarized into three overall components:

- Set up an ordinary kriging model for a given case area.
- Create a parametric emulator for fast kriging computations.
- Use the emulator in a fast optimization procedure to find optimal locations for new rain gauges.

Figure 2 shows a flow diagram of how each step fits into the framework. The individual subsections of the methods chapter will provide a detailed explanation for each step of the process.
Figure 2: Flow diagram of the developed framework. Each step in the framework is here shown as part of the three main components: the ordinary kriging framework, the emulator and the optimization procedure.

3.1. Kriging-based spatial interpolation

The current IDW-algorithm used by DMI to interpolate annual precipitation is deterministic and therefore does not provide any estimation of the uncertainty regarding the predicted values. Instead we used the geostatistical interpolation method ordinary kriging, which also includes the variance of the interpolated values. The choice of ordinary kriging over other methods is also motivated by its long-standing and still frequent use for this kind of rain gauge network design exercise in the scientific literature (Adhikary et al., 2015; Attar et al., 2019; Bayat et al., 2019, 2021b; Chen et al., 2008; Pardo-Igúzquiza, 1998; Xu et al., 2018).
3.1.1. Variogram modelling

Kriging relies on the variation in the rain data being correlated. Like most other regionalized variables, the similarity in measurements tends to be closer in sample points close to each other than ones further away. To describe these relationships the variance-covariance matrix is used, and within kriging-frameworks the underlying variogram can be used to build it. The relation between the variogram $\gamma(h)$ and the covariance function $C(h)$ is given by

$$\gamma(h) = C(0) - C(h),$$  \hspace{1cm} (1)

where $h_{i,j} = x_i - x_j$ (lag vector between sample points $x_i$ and $x_j$). The true variogram is of course not known and has to be estimated by first deriving the empirical variogram $\gamma^*(h)$ from the data, which is then used to fit the parameters in a variogram model (Webster and Oliver, 2007). To derive the empirical variogram, the pairs of data points are divided into classes so that all pairs of points with a distance $|h|$ within a certain interval belong to the same class. Doing so relies on an assumption of isotropy, i.e. that the properties of the variogram is only dependent on the distance from a sample point and independent of the direction of movement away from the point. As an example we could divide our pairs into intervals of 5 km, then the classes would be defined as $H_k := \{h \in \mathbb{R}^2 : (k - 1) \cdot 5km \leq |h| < k \cdot 5km\}$ where $k \in \{1, 2, \ldots, K\}$. After determining the classes, the average dissimilarity for each class is calculated with the method-of-moments resulting in the empirical variogram

$$\gamma^*(H_k) := \frac{1}{2|N(H_k)|} \sum_{N(H_k)} (z(x_i) - z(x_j))^2, \hspace{0.5cm} k \in \mathbb{N}.$$  \hspace{1cm} (2)
Here $N(H_k) = \{ x_i - x_j \in H_k \ for \ i,j = 1,\ldots,n \}$ and $|N(H_k)|$ is the distinct number of elements in $N(H_k)$, while $z$ is the data value (e.g. rainfall). An example of an empirical variogram can be seen in Figure 3.

![Variogram Example](image)

**Figure 3:** Example of an experimental variogram where the stars represent the dissimilarity in a binned variogram with 50 bins. Each star represents the mean of a bin. The figure also serves as an illustrative example of the classic geostatistical parameters: range, sill and nugget.

The three most important parameters for the variogram are the **nugget**, **sill** and **range** illustrated in Figure 3 and which are defined as:

1. Nugget: The value of $\gamma(h)$ at an infinitesimally small distance from a sample location, defined as $\gamma(h) \rightarrow c_0 > 0$ as $|h| \rightarrow 0$, where $c_0$ denotes the nugget.

2. Sill: The value of $\gamma(h)$ at the point where the semivariogram levels off, defined as $\gamma(\infty) = \lim_{|h| \rightarrow \infty} \gamma(h)$. 
3. Range: The distance $|h|$ after which the semivariogram $\gamma(h)$ levels off.

The computed empirical variogram can be used to fit a parametric theoretical variogram model by estimating the nugget, sill and range. Some models frequently used in hydrology include the Spherical model, the Exponential model and the Gaussian model. We considered all three with an added nugget-effect when determining the best fit for our data.

When determining the best fit, we considered the root mean squared error (RMSE) between the variogram model and the empirical variogram. Since we have DMI’s interpolated maps available to us, which act as the official precipitation maps of the Danish government and are developed by local experts in the field, we also considered the deviation between the interpolation resulting from the variogram and DMI’s IDW-based interpolation. By fitting the model against the IDW-maps, the results of this study will on top of the novel methodological developments also have practical relevance for other studies that employ this precipitation data sets as well as stakeholder discussions about the future of Danish climatological monitoring of precipitation.

Since DMI’s IDW model considers coastal-inland climate classification, we have done the same by adding them to the distance function, when creating the empirical variogram. The distance between two points $x_i, x_j$, is computed by

$$
\text{dist}(x_i, x_j) = \sqrt{V_{\text{Vin}}(x_i, x_j)^2 + w_{\text{climate}} \cdot (\text{ClimateVal}_{x_i} - \text{ClimateVal}_{x_j})^2}
$$

(3)

Here $V_{\text{Vin}}(x_i, x_j)$ is the Vincenty distance (Vincenty, 1975), $w_{\text{climate}}$ is the
weight applied to the climate values, and \( \text{ClimateVal}_{x_i} \) is the specific climate value of point \( x_i \). Notice that \( w_{\text{climate}} \) determines the degree to which local differences in climate values impact the spatial interpolation, and setting it to 0 will result in the standard distance measure. The value of \( w_{\text{climate}} \) is estimated by minimizing the RMSE between the resulting interpolation and DMI’s IDW product. When changing the distance function, one have to be careful to stay in a metric space, to keep the variogram models well-defined. That is, the defined distance function have to be a norm, i.e. be positive definite, have absolute homogeneity and be subadditive. In our case we still have a metric space as our defined distance function is a product metric.

The empirical variogram will vary between years and to ensure a more robust estimation of the variogram models, we utilized data for several consecutive years (2016-2021). Once a final variogram model has been selected, we want to validate that it provides reasonable interpolations not just for average rainfall over several years but also for individual years in the data set. This is done by calculating the RMSE between the kriged maps and DMI’s IDW map. This was done both for the variogram model trained over the entire time period (2016-2021) and variogram models trained for each individual year. To validate that the prediction error variance is well-calibrated, a leave-one-out cross validation was performed where individual rain gauges were left out on by one. It was then checked how many of the left out observations were within the 95% interval, which ideally should be a number close to 95%.

### 3.1.2. Ordinary kriging

Ordinary kriging is a geostatistical method of spatial interpolation. In ordinary kriging, the estimated value \( Z^*(x_0) \) at an unsampled point \( x_0 \) is
computed by a linear (weighted) combination of realized values from the sample points ($S$).

$$Z^* (x_0) := \sum_{i=1}^{n} \omega_i Z(x_i) = \omega^T Z, \quad x_i \neq 0 \in S,$$  \hfill (4)

where $\omega := (\omega_1, \ldots, \omega_n)^T \in R^n$ are the weights. The prediction error variance can be computed by

$$\sigma^2 = \lambda - \omega^T C_0 + C(0),$$  \hfill (5)

where $C_0 = (C(x_1 - x_0), \ldots, C(x_i - x_0))^T \in R^n$, and $\lambda$ are Lagrange multipliers. The optimal weights are those that minimize the prediction error variance in order to ensure precise predictions. The optimal weights are computed by

$$\begin{pmatrix} \omega \\ \lambda \end{pmatrix} = \tilde{\Sigma}^{-1} \tilde{C}_0,$$  \hfill (6)

where $\Sigma$ is the covariance matrix and

$$\tilde{\Sigma} = \begin{pmatrix} \Sigma & 1 \\ 1^T & 0 \end{pmatrix}, \quad \text{and} \quad \tilde{C}_0 = \begin{pmatrix} C_0 \\ 1 \end{pmatrix}.$$  \hfill (7)

3.2. Emulator model for fast kriging computations

The size of the covariance matrix in a kriging setup is directly related to the number of rain gauges in the full network. Large rain gauge networks lead to large covariance matrices and inverting these as done in Equation 6 can be very computationally expensive. The aim is therefore to develop a computationally cheaper method for approximating the results that would come out of placing a new gauge on the map and once again perform a full kriging interpolation. In the rest of the paper, we refer to the computationally
cheaper method as the kriging "emulator". Setting up this emulator requires three steps:

1. Selecting a set of trial points within the case area that will form the training data for the emulator.
2. Defining an appropriate emulator model that can approximate the local effects on the kriging uncertainty map from placing a new gauge in one of the trial points.
3. Validating the outputs of the emulator model.

3.2.1. Selecting a set of trial points

The first step in creating the kriging emulator is to select a set of trial points on the uncertainty map generated by the ordinary kriging algorithm. It is important to choose points that represent the variations on the map by sampling locations with low and high variance. The number of points necessary to obtain an accurate emulator model is a tuneable hyperparameter and for the case of Denmark we found that seven points performed well. The seven trial points are shown on Figure 4.
Figure 4: Location of the selected trial points (blue dots) used to create a kriging emulator. The points are placed such that a broad spectrum of local conditions are represented on the original ordinary kriging uncertainty map (as displayed by the color scale of the background map).

For each individual trial point, a new uncertainty map resulting from placing a rain gauge at the point is computed with ordinary kriging. With seven points this leads to seven new uncertainty maps. Placing a new gauge naturally leads to a reduction in the interpolation uncertainty in the area surrounding the trial point. For each grid cell within the full case area in each of the seven new maps, the reduction in standard deviation is recorded (we will refer to this as the "sd gain"). The emulator’s objective is to accurately
predict this sd gain in a fast manner without having to perform ordinary kriging.

3.2.2. Defining and estimating a suitable emulator model

To provide fast computation of the sd gain, we propose a relatively simple parametric function that takes three variables as inputs:

- The distance between the candidate location of a new rain gauge and the grid cell for which we want to compute the gain.
- The standard deviation above the square root of the nugget in the grid cell for which we want the gain.
- The standard deviation above the square root of the nugget in the point where the new gauge is posed (before it is inserted).

Notice that none of these variables require more than the original kriged map to be computed. The parametric formulation of the emulator model should reflect the shape of the sd gain as a function of the input variables. When a formulation has been settled upon, its parameters can be fitted using the least squares method. For each trial point in Figure 4, the three variables were recorded for the rest of the points in the entire grid. From this we created a point cloud as seen in Figure 5, on which we wished to fit our parametric function.
Figure 5: The gain in kriging standard deviation as a function of distance from proposed new gauge location and current standard deviation at the grid cell (a). A side-view of (a) to highlight the relation between distance from trial point and the resulting gain (b). The different colors represent data from each of the trial points in Figure 4, while each dot represents the gain in a single grid cell close to the location for the new gauge.

From analyzing the point cloud in Figure 5 it is clear that observations do not fall on a single surface, as e.g. the orange points are clearly below some of the yellow, purple, and green clouds. However, inspecting the individual clouds reveals that the clouds almost perfectly fall on a surface when restricted to one trial point. Furthermore, it seems that some kind of exponential decay is present over the distance, and similarly that some sort of decay is present negatively correlated with the standard deviation above the nugget (see Figure 5(b)). Finally, it might seem like the points in the plane with distance from the trial point equals 0 fall on what might be a logarithmic line. Using our observations we pose a parametric function $f(x, y, z)$ to be fitted to the point cloud

$$f(x, y, z) = a_1 \log(a_2 z + 1)e^{-c x z - y + b_2 (z + b_1)^d},$$  

(8)
where \( x \) is the distance from the new station, \( y \) is the standard deviation above the square root of the nugget for a grid cell, and \( z \) is the standard deviation above the square root of the nugget at the location of the new station, and all parameters \((a_1, a_2, c, b_1, b_2, d)\) are greater than 0. With this function we capture the logarithmic nature, when the distance is 0 since \( f(0, y, z) = a_1 \log (a_2 z + 1)e^0 = a_1 \log (a_2 z + 1) \). Furthermore we capture the exponential nature seen in Figure 5(b) when the distance \( (x) \) grows.

3.2.3. Validating the emulator

Having fitted the function to the data, we can proceed to test if the function is indeed representative of the kriging process. We tested the predictions made by our kriging emulator by computing the kriged effect on the average standard deviation by placing a gauge in a specific point, for a subset of test point (or the whole grid). The structure of the residuals between the emulator predictions and the kriged map were visually examined to make sure that there were no significant processes missing in the emulator. The relative size between the absolute values of the residuals were also compared to the values of the original kriged map to make sure that they were deemed low enough for the intended use of the emulator.

Note that this method is very computationally efficient. The heaviest computations are the kriging process and simply fitting the parameters in the parametric function. Let \( j \) be the number of trial points (in our case \( j = 7 \)), the user then has to estimate the variogram once, compute the kriging uncertainty map \( j + 2 \) times, and fit the parametric function to the \( j \) point clouds. The computational aspect only has to be done once and takes in the order of minutes to compute.
Defining the exact form of the emulator should be seen as an iterative process. If the predictions made by the emulator does not resemble the actual kriging process, we can go back to the phase of creating the training set or analysing the point cloud. In the creation of the training set we can add more trial points or move them around. In the analyzing phase we can change our custom parametric function, so that it better represents the data. When the predictions made by the estimated emulator resembles the actual kriging process to a satisfactory degree, then the emulator can be used for the optimization heuristic, making it many times faster than if kriging was utilized.

3.3. Heuristic for placing new gauges

In the case study for this paper, we considered the centroid of each of the 46,797 cells in the 1x1km² grid over Denmark as a potential new location for a rain gauge. In an example where 150 new gauges has to be placed, an optimization procedure would have to deal with a combinatorial n-choose-k (46,797-choose-150) problem with $4.7 \cdot 10^{137}$ potential solutions. To deal with such a large solution space, we propose a new optimization heuristic based on sequential optimal placements of gauges followed by a final resubstitution step to mitigate some of the problems sequential placement might lead to.

3.3.1. The cost function

First the cost function for the optimization is defined, where the goal is to maximize the coverage and precision of the rain gauge network. This will be measured by the average standard deviation across the whole case area. Let $N$ be the number of points in the grid, the cost function to be minimized
is then
\[ S(x_1, \ldots, x_k) = \frac{1}{N} \sum_{i=1}^{N} \sigma_i(x_1, \ldots, x_k). \] (9)

Here \( k \) is the number of gauges to be placed and \((x_1, \ldots, x_k)\) are the positions of these new gauges. If any areas of the map are of particular interest one can add a weight term to the cost function, weighting each grid cell. Alternatively, if an area is not of interest the corresponding weight could be set to 0.

3.3.2. Placing new gauges sequentially

The optimization heuristic requires the user to have already found an underlying variogram for the area of interest and trained an emulator model. It then works by going through the following steps:

1. Compute a kriging uncertainty map for the full case area.
2. Consider every location in the grid without a rain gauge as a candidate for the placement of a new gauge (in the presented case study there are 46,797 grid cells over Denmark, which is therefore the number of potential locations to consider).
3. Loop through each candidate and use the fast emulator to compute a new (emulated) kriging uncertainty map for a network that includes the candidate gauge.
4. Estimate the cost function for including a gauge at each candidate location, and choose the location with the best result.

Steps 1-4 is repeated \( k \) times, where \( k \) is the number of gauges one wish to place.
3.3.3. Dealing with the potential problems of sequential placement

The fact that new gauges are inserted sequentially (i.e. greedily) means that the first selected locations will have an effect on the choice of later selected locations when placing many new gauges. To demonstrate, consider the following constructed example in Figure 6, where three new gauges are placed in the triangular area. In the example, the exact parameters for the kriging are not essential, nor are the values in the uncertainty map. The critical feature here is that a greedy algorithm would most likely place the first gauge right at the center of the triangle with the highest uncertainty, followed by two gauges at two of the corners like in Figure 6(b). It is, however, likely that the positioning in Figure 6(c) is a better result since it is more symmetrical and also what we would intuitively expect by looking at Figure 6(a). Thus, the heuristic should also be able to produce the result in Figure 6(c), if it is indeed a better solution. To obtain this, a relocation process is introduced where each of the gauges are sequentially removed and reinserted in the order the gauges were placed, starting by removing the gauge that was placed first. We refer to one iteration of this relocating process as a "resubstitution". With this method, the hope is that when removing the gauge in the center of the triangle, the heuristic sees that it is better to place it a little more towards the uncovered corner. Of course, this process effectively multiplies the expected running time of the heuristic by the number of resubstitutions. In total, the accumulated running time is expected to be of order $O(k^3)$ due to matrix inversion where $k$ is the number of new stations. However, our experiments have shown that in practice, this is more comparable to $O(k^2)$ behaviour as the heuristic takes $O(k)$ to place
the k'th new station while the time for the kriging after (where the matrix inversion happens) is somewhat constant. So the heuristic is expected to be a much faster alternative to solving the problem from scratch through a genetic optimization algorithm, where one would gradually evolve a population of potential solutions closer to the optimum in an analogous way to how natural selection evolves over time. An example of such a setup used for rain gauge network design is the NSGA-II algorithm implemented by Wang et al. (2020), which requires thousands of simulations to search through solution space of $6.7 \cdot 10^7$ combinations, which is much smaller than the space considered in this study.
Figure 6: A constructed example map of a somewhat trivial scenario for where three new stations are to be placed (a). Results of a sequential, greedy heuristic placing 3 gauges (b). A more symmetric and (possibly) better solution (c).

4. Results and discussion

4.1. Variogram model

When fitting the parametric variogram model against the empirical variogram, we found that weighting each bin according to the number of observations in it yielded a lower RMSE when compared to DMI’s IDW map than weighting each bin uniformly irrespective of the number of observation
in them. Of the three tested variogram models, the spherical and exponential variograms performed much better than the Gaussian model in terms of the RMSE against the empirical variogram as well as the RMSE between the resulting rainfall interpolations and DMIs interpolated map (not shown). The spherical model performed slightly better than the exponential fit and is therefore used for the remainder of the results. The spherical model with a nugget effect and the corresponding covariance function are given by

\[
\gamma_{a,c,c_0}(h) := \begin{cases} 
0, & \text{if } |h| = 0 \\
 c_0 + c \left(\frac{3|h|}{a} - \frac{1}{2} \left(\frac{|h|}{a}\right)^3\right), & \text{if } 0 < |h| \leq a \\
 c_0 + c, & \text{otherwise},
\end{cases} 
\]

\[
C_{a,c,c_0}(h) := \begin{cases} 
 c + c_0, & \text{if } |h| = 0 \\
 c \left(1 - \frac{3|h|}{a} + \frac{1}{2} \left(\frac{|h|}{a}\right)^3\right), & \text{if } 0 < |h| \leq a \\
 0, & \text{otherwise},
\end{cases} 
\]

where \(a\) is the range, \(c\) is the partial sill, \(c_0\) the nugget, and \(c + c_0\) the sill. The optimal values for the three parameters were estimated to \(a = 7.5 \cdot 10^5 m\), \(b = 5.2 \cdot 10^4 mm^2\) and \(c_0 = 1.7 \cdot 10^3 mm^2\). The optimal weighting of the climate values was found to be \(w_{\text{climate}} = 300m\). The resulting variogram is visualized in Figure 7(a). Notice that the variogram fit is much closer to the first half of the bins at the expense of a relatively poorer fit for bins at longer distances. This is a result of the way the weighting of the bins was done, as the first half of the bins contain a lot more observations than the latter bins.

The variogram model presented in Figure 7(a) was trained on all years simultaneously (2016-2021), and this variogram model will be used for the
results in the following sections. The results of this study have practical implications for the design of Denmark’s gauging network in addition to the methodological developments, and we are therefore interested in evaluating the realism of the model’s kriged rainfall estimates and error variance. To validate that the model provides realistic and robust rainfall interpolations, it is interesting to compare it to the optimal variogram models trained on each individual year, since rainfall can be highly variable across time. Variogram models trained on data from single years are shown in Figure 7(b)), where it is seen that the variograms of individual years are quite different. We compare the performance of the chosen model’s kriged rainfall maps against the models shown in Figure 7(b) by calculating the RMSE between the kriged maps and the main stakeholder’s (DMI’s) current IDW-based maps for each individual year. From Table 1 it is seen that the chosen variogram model actually performs very well on the individual years. For 2016, 2017, 2019, and 2021, it even performs better than the models trained using only data from the respective years. This result increases the confidence in the chosen variogram model as we use it for network design.
Figure 7: Empirical variogram and estimated fit using data from 2016-2021 (a). Individual variogram fits for each year trained with data from only the corresponding year (b).
Table 1: RMSE between DMI’s interpolated map and the kriged interpolations using data from the individual year and using data from all the years. For comparison the minimum obtainable standard deviation is $\sqrt{c_0} = 41.3\,\text{mm}$.

<table>
<thead>
<tr>
<th>Year</th>
<th>Models trained on individual years [mm]</th>
<th>Model trained on all years [mm]</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021</td>
<td>30.7</td>
<td>30.2</td>
</tr>
<tr>
<td>2020</td>
<td>28.4</td>
<td>28.7</td>
</tr>
<tr>
<td>2019</td>
<td>35.3</td>
<td>35.1</td>
</tr>
<tr>
<td>2018</td>
<td>17.7</td>
<td>19.4</td>
</tr>
<tr>
<td>2017</td>
<td>34.1</td>
<td>33.1</td>
</tr>
<tr>
<td>2016</td>
<td>30.1</td>
<td>28.9</td>
</tr>
</tbody>
</table>

Since not only the interpolated values but also the kriging prediction uncertainty is important for the developed method, a leave one out cross validation on the 95% prediction interval was also performed. Individual rain gauges were left out one by one, and it was then checked if the observed rainfall of the removed gauge were within the prediction interval of the chosen variogram model from Figure 7(a). Table 2 summarizes this experiment on a year by year basis. It is seen that close to 95% of the observations lie within the 95% prediction interval in all years, which means that the prediction uncertainty of the chosen model is well-calibrated.
Table 2: Percentage of observed values that fall within the 95% prediction interval of the kriging uncertainty map generated by the chosen variogram model.

<table>
<thead>
<tr>
<th>Year</th>
<th>Within 95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>2021</td>
<td>95.1%</td>
</tr>
<tr>
<td>2020</td>
<td>95.2%</td>
</tr>
<tr>
<td>2019</td>
<td>92.9%</td>
</tr>
<tr>
<td>2018</td>
<td>98.6%</td>
</tr>
<tr>
<td>2017</td>
<td>95.9%</td>
</tr>
<tr>
<td>2016</td>
<td>96.3%</td>
</tr>
</tbody>
</table>

With the variogram model considered validated, it was possible to produce a kriging uncertainty map for the rain gauge network configuration in 2022 (see Figure 8), which can then be used to train the emulator model.
Figure 8: Kriging uncertainty map for the gauging network configuration in the year 2022. White dots are locations of rain gauges. The minimum obtainable standard deviation is $\sqrt{c_0} = 41.3$ mm.

It is worth noting that the spatial correlation of rainfall, and thus also the variograms, is dependent on the temporal aggregation of the data. In this study, we have used annual rainfall sums for the network design process, but different network configuration could be generated with daily or monthly aggregation levels.
4.2. Kriging emulator

The parameters in the emulator were trained on data from the seven trial points shown in Figure 4, which were sampled to represent areas of low, medium and high kriging standard deviations in the original kriging uncertainty map. The goal of the emulator is to predict the degree to which placing a new gauge in a specific location improves the kriging uncertainty in nearby grid cells. Figure 9 shows the emulator predictions against its training data as well as the resulting residuals for three of the trial points (one low uncertainty, one medium uncertainty, one high uncertainty). In Figure 9(a, c, e) it is seen that the surfaces resulting from emulator predictions of the gain in kriging standard deviation fit the observed gains well. When inspecting the residual plots (Figure 9(b, d, f)) it is seen that the residuals are not simply white noise, since there is some left over structure. This indicates that there are some external effects the emulator does not account for. One such effect could likely come from how the location of other existing gauges in the network influence the gains from placing a new gauge in a given location. Another effect might be local influences of the coast-inland climate values. The emulator indirectly accounts for this in a relatively simple way by taking the current standard deviation values in the grid as an input in the form of the $y$-parameter in Equation 8. The residuals are, however, rather small.

To further test the quality of the emulator, it was applied to the full case area and compared to the original, computationally expensive ordinary kriging. Figure 10(a) shows the emulator estimates of gains on the national average kriging standard deviation (“sd gain”) from placing a new gauge at that location. Thus placing a gauge in the areas with the highest gain would
be most beneficial to the overall objective. Figure 10(b) shows the same gains computed with the ordinary kriging algorithm, while Figure 10(c) shows the absolute difference between the two maps. The emulator predictions are generally very close to the original kriging estimates with the largest absolute errors in Figure 10(c) being around 0.0075 mm/year.

Creating the maps in Figure 10(a) and (b) required 46,797 simulations of inserting a single gauge in each potential location and evaluating the cost function. Running 46,797 simulations with the emulator took under 5 seconds, while 46,797 kriging runs in Figure 10(b) took approximately 4 hours on a regular laptop (Asus Zenbook, 16 GB ram, 8 AMD Ryzen 7 cores). That is a speed-up close to a factor of 3000. Keep in mind that this map has to be created once for each new gauge that has to be placed during the optimization procedure. This per-gauge speed-up from a scale of hours to seconds is what enables us to even consider optimization on a large rain gauge network.
Figure 9: The blue dots show observed gains in kriging standard deviation from inserting a gauge at the trial points vs. the emulator predictions as a red surface (a, c, e). Residuals between emulator predictions and observations (b, d, f). The three rows show data from a trial point with low sd (a, b), medium sd (c, d), and high sd (e, f).
4.3. Application on a large rain gauge network

With the optimization heuristic procedure explained in Section 3.3 and the cost function shown in Equation 9 we tried placing 175 new gauges as shown in Figure 11(a). Here it is seen that the new rain gauges were placed in intuitively reasonable areas that previously had a high kriging uncertainty.
(Figure 8), while the country’s urban areas that already were well-monitored do not receive new gauges.

Figure 11(c) shows how the national average kriging standard deviation as well as the 90th percentile on the map was reduced as a function of the number of new gauges. This information highlights the diminishing marginal return of continuing to install new gauges, as the first new gauges closed many of the largest gaps on the map. This information can be used by stakeholders to decide how large of an investment they want to make relative to improvements in national rainfall monitoring. Note that the optimization preferred placing new gauges on the large Jutland peninsula and the largest islands of Denmark, because the applied objective function was the national average standard deviation. That also means that even though 175 new gauges were placed on the map there were still minor islands that did not receive a gauge and thus remain with high kriging uncertainty (see the small high uncertainty dots in the white ocean areas in Figure 11(a, b)). If it would be desired to also monitor these islands (or other specific areas of interest), it would be relatively easy to work in a weighting of these locations in the objective function.

Figure 11(d) shows the effect of using multiple resubstitution rounds to mitigate some of the issues with sequential gauge placements. From the figure it is clear that resubstitution only provided small benefits when placing few gauges. That is because the greedy nature of the algorithm picked the most high-value locations first, and these were spread out throughout the area. The benefit of resubstitution started to show once several new gauges were placed in near proximity to each other, since this was when effects of
previously selected locations would impact the selection of the next locations. That also explains why Figure 11(d) shows that the benefit of performing resubstitution continued to increase with the number of new gauges that had to be placed. Figure 11(d) also indicates that simply performing one round of resubstitution for this case study provided most of the benefit one might obtain from continuing with several rounds. As an example of the resubstitution effect, Figure 11(b) shows how the gauges moved compared to Figure 11(a) when three resubstitutions were made. The overall spatial distribution of new gauge locations were similar, but there were many small local changes.
Figure 11: The white dots represent locations for the 175 new gauges from using the heuristic without resubstitution (a). The locations of 175 new stations using heuristic resubstitution three times (b). Average national standard deviation (the objective function) and the 90th percentile as a function of number of stations placed (c). Change in average standard deviation when using 1-7 resubstitutions as a function of number of new gauges placed (d).

Running the optimization with the emulator once took approximately 15 minutes (Figure 11(a)). Performing resubstitution three times increased the run time to roughly one hour (Figure 11(b)). We did not run the optimization heuristic with the original kriging procedure instead of the emulator (since that would be very demanding), but extrapolating the x3000 speed-up for this
case would mean that a kriging-based optimization would take a prohibitive 125 days to complete. This clearly highlights how the presented emulator in combination with the optimization heuristic enables large-scale rain gauge network augmentation efforts, which were previously not possible.

It should be noted that the proposed methods are not unique to the problem of optimizing a rain gauge networks, as they can be applied to any network of spatial data where one is able to apply ordinary kriging as part of an optimization framework.

4.4. Final comments on the emulator-based framework

In this study, ordinary kriging was used as the reference method for deriving the geostatistical interpolations. In this setting the covariance functions rely on covariance between points on a grid, which implies that the emulator works on residual covariance matrices. Universal kriging is another geostatistical method and can include a number of covariates that depend on the specific location of each point, such as elevation or land use. Such a model may change the covariance structure of the residuals as the trend coefficients in that model also has to be estimated. The principles presented in this article should, however, be the same; namely to find the best emulator relative to the residuals. While the exact form of the emulator may change, the overall principle remains the same.

The aim of the emulator is to approximate the results of the ordinary kriging setup. The quality of the approximation will determine how much uncertainty the emulator causes in the final optimized network design. There are three places in the estimation of the proposed emulator where uncertainty can arise: (1) in the choice of structure for the parametric function, (2) in
the creation of training data, and (3) during estimation of the parameters in
the function. Any error that arise from these three steps will induce a bias
in the kriging estimates. It is therefore important to check that the emulator
approximates the kriging estimates to a satisfactory degree for the specific
use case.

As a last comment, the timings of computational efficiency presented in
this article are of course dependent on the specific code implementation made
of both the emulator and the kriging algorithm. In addition to the novel em-
ulator and optimization heuristic, we also coded the ordinary kriging imple-
mentation in Python ourselves. Hence there might be specific computational
matrix tricks (or other improvements) that can improve calculation speed for
both the kriging process and the emulator, which we have not implemented.
An example being that when a station is added to an existing gauging net-
work of size \( n \), the \( n \times n \) submatrix of the \((n+1) \times (n+1)\) inverted covariance
matrix is already known and thus does not need to be computed again.

5. Conclusion

This study presents a workflow with high computational efficiency that
enables optimal placements of many new rain gauges in a classic rain gauge
network augmentation task. The main novel contributions of the paper are
the development of an efficient model that emulates a traditional ordinary
kriging approach, and a greedy optimization heuristic with a resubstitution
mechanism.

The proposed emulator is an efficient and parsimonious parametric func-
tion that approximates the kriging uncertainty improvements from adding
a new gauge within the case area. The user only needs to specify and estimate a variogram model for a case area before the method can be used. The emulator takes three input variables, which is given by performing kriging once with the estimated variogram: (1) the distance between the proposed location of a new gauge and a given grid cell within the case area, (2) the current kriging uncertainty in the given grid cell, and (3) the current kriging uncertainty in the proposed gauge location. The results showed that the emulator was able to approximate the original ordinary kriging well (residuals less than 0.0075 mm/year) and with a computational speed-up of a factor of 3000.

With the computationally cheap emulator, it was possible to develop a greedy heuristic for placing many new gauges in a large optimization problem. The heuristic imposes a grid over the case area and considers all cells currently without a gauge as a candidate location for a new gauge. The optimization then uses the emulator to explore what the improvements of placing a new gauge in all candidate locations will be, chooses the location with the largest benefit, and repeats this for the number of new gauges that the user wants added. This type of sequential placement of new gauges can lead to sub-optimal solutions and to combat this a resubstitution mechanism is introduced. The resubstitution loops through the proposed locations and explores if another location has become a better choice given the effect of other new gauges being placed in the vicinity. The effect of the resubstitution mechanism was tested on the case area and it was clear that the necessity of this step increases with the number of new gauges that needs to be evaluated.
The workflow was tested on a large gauging network in Denmark with 291,707 existing rain gauges. By applying the emulator and optimization heuristic it was possible to optimally locate 175 additional rain gauges for the network in roughly 15 minutes plus 15 minutes for each round of resubstitution. The equivalent exercise would have taken a prohibitive 125 days to compute with a traditional ordinary kriging approach. This shows the clear potential of the developed framework to aid stakeholders in handling complex, large rain gauge network augmentations.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Jonas Wied Pedersen was financially supported by the Innovation Fund Denmark through the industrial postdoc program [grant number 0197-00005B].
References


Declaration of interests

☑ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: