

Evaluation of Bio-Inspired Algorithms in Cluster-Based Kriging Optimization

Carlos Yasojima(✉), Tamara Ramos, Tiago Araujo, Bianchi Meiguins, Nelson Neto, and Jefferson Morais

Postgraduate Program of Computer Science, Federal University of Pará, Belém, Pará, Brazil
takeshiyasojima@gmail.com, tamarasabrininew@gmail.com, {tiagoaraujo, nelsonneto, jmorais}@ufpa.br

Abstract. Kriging is one of the most used spatial estimation methods in real-world applications. In kriging estimation, some parameters must be estimated in order to reach a good accuracy in the interpolation process, however, this step is still a challenge. Various optimization methods have been tested to find good parameters to this process, however, in recent years, many authors are using bio-inspired techniques and reaching good results in estimating these parameters. This paper presents a comparison between well-known bio-inspired techniques such as Genetic Algorithms, Differential Evolution and Particle Swarm Optimization in the estimation of the essential kriging parameters: nugget, sill, range, angle, and factor. We also proposed an improved cluster-based kriging method to perform the tests. The results shows that the algorithms have a similar accuracy in estimating these parameters, and the number of clusters have a high impact on the results.

Keywords: Bio-inspired algorithms · Artificial Intelligence · Geostatistic · Kriging

1 Introduction

Kriging is a geostatistical interpolation technique that predicts the value of observations in unknown locations based on previously collected data. The kriging error or interpolation error is minimized by studying and modeling the spatial distribution of points already obtained. This spatial distribution or spatial variation is expressed in the form of an experimental variogram [1].

The experimental variogram can be considered as a graphical representation of the data distribution and also expresses the data variance with the increment of the sampling distance. The variogram is the basis for the application of the kriging method. Thus, the kriging process is defined in three main steps. First, the experimental variogram is calculated. Then, the theoretical variogram is modeled to represent the experimental variogram. Finally, the value of a given point is predicted using the built theoretical model.

© Springer Nature Switzerland AG 2019
S. Misra et al. (Eds.): ICCSA 2019, LNCS 11619, pp. 731–744, 2019.
https://doi.org/10.1007/978-3-030-24289-3_54

The theoretical variogram can be modeled, for example, by estimating some parameters: nugget, sill, range, factor, and angle in anisotropy problems, where the spatial behavior is different in distinct directions.

Artificial Intelligence Techniques have been used with kriging technique to produce better results as shown in [2–10], whereas it is still a challenge to determine what method is better structured for a database.

As stated in [7] and applied in [8], bio-inspired algorithms, in general, are suited to help define the theoretical variogram parameters. Furthermore, these types of algorithms do not require a single initial value as input, but rather an interval. In [3], the authors achieved good results adopting Genetic algorithm technique.

Various researches have been implementing bio-inspired algorithms to optimize these parameters, such as Genetic Algorithms (GA) [6, 8, 9], Differential Evolution (DE) [10, 11] and more recently PSO [12]. However, it is important to compare the behavior of these algorithms when applied to the same problem and study their effectiveness in optimizing variogram parameters.

In order to evaluate these algorithms, it is important to select the method to evaluate the goodness of each solution. In [8], it is shown that the interpolation cost function, which we use as the fitness function to evaluate the set of parameters, generally yield better results at the expense of computational cost. Well-known cost functions, such as Weighted Least Squares, Maximum Likelihood and variations were not considered in this paper.

To apply state-of-art kriging methodologies, we use Cluster-Kriging method proposed in [13]. In this method, the spatial data are divided into different sub-groups, where each data is interpolated using only data from same group. A limitation of Abedini methodology is a single variogram for all groups, we propose the estimation and optimization of different parameters to each group. Some problems of this improvement are solved using K-Nearest Neighbour method and data preprocessing as shown in the next section.

In this context, the purpose of this paper is to evaluate well-known bio-inspired techniques such as Genetic Algorithms, Differential Evolution and Particle Swarm Optimization in the estimation of the essential kriging parameters. We also proposed an improved cluster-based kriging method to perform the tests. The results show that the algorithms have a similar performance, but the clustering algorithm and the number of clusters might have a high impact on the kriging results.

This paper is organized as follows. Section 2 consists of the theoretical background involving important concepts for the understanding of this paper. In Sect. 3, the steps of the proposed method are detailed. Section 4 presents the databases used in this work besides the results of the experiments performed. Finally, Sect. 5 presents final considerations.

2 Background

This section presents a brief explanation of the main concepts discussed in this paper.

2.1 Kriging

Let Z be a set of measurements of a target variable denoted as $\{z(s_1), z(s_2), \ldots, z(s_N)\}$, where $s_i = (x_i, y_i)$ is a data point in a spatial database; x_i and y_i are its coordinates (primary locations); and N is the number of observations.

Values of the target variable at some new location s_0 can be calculated using a spatial prediction model. The standard version of kriging is called ordinary kriging (OK), where the predictions are based on the model:

$$\hat{z}_{OK}(s_0) = \sum_{i=1}^{n} w_i(s_0).z(s_i) = \lambda_0^T \cdot \mathbf{z} \tag{1}$$

where λ_0 is a vector of kriging weights (w_i), and z is the vector of N observations at primary locations.

So, to estimate the weights, we calculate the semivariances $\gamma(h)$ based on the differences between the neighboring values:

$$\gamma(h) = \frac{1}{2} E[(z(s_i) - z(s_i + h))^2] \tag{2}$$

where $z(s_i)$ is the observation of the target variable at some point location, and $z(s_i + h)$ is the observation of the neighbour at a distance $s_i + h$.

Suppose that there are N point observations, this yields $Nx(N-2)/2$ pairs for which a semivariance can be calculated. If we plot all semivariances versus their separation distances a variogram cloud is produced. For an easier visualization of this variogram cloud, the values are commonly averaged for a standard distance called "lag". If we display such averaged data, then we get the standard experimental variogram, which can be seen in Fig. 1.

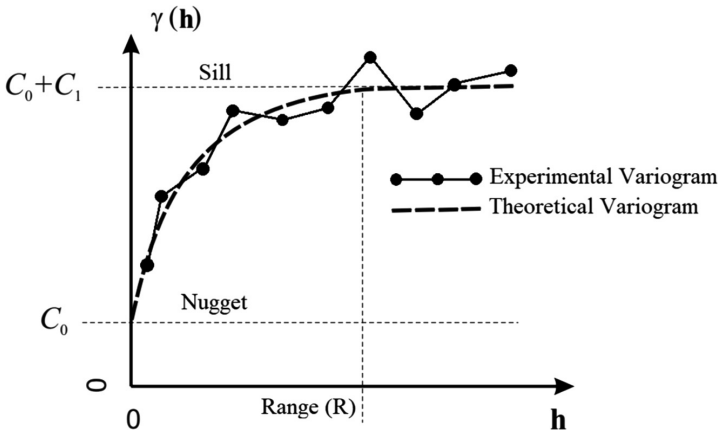

Fig. 1. Kriging variogram and parameters.

Once we calculate the experimental variogram, we can fit it using a theoretical model, such as linear, spherical, exponential, gaussian, among others. The variograms are commonly fitted using a cost function (e.g. weighted least squares [15]). Hence, the main objective is to minimize this cost function. The theoretical model used in this work is the exponential one, given by

$$C(h) = \begin{pmatrix} C_0 + C_1 & |h| = 0 \\ C_1.[e^{-(\frac{h}{R})}] & |h| > 0 \end{pmatrix} \tag{3}$$

where R is the practical range, which is the coordinate where the model starts to flatten out.

C_0 is the "nugget effect" that is attributed to measurements errors or spatial sources of variation at distances smaller than the sampling interval. $C_0 + C_1$ is the sill, which is the value that the model attains at the practical range. These parameters, also called coefficients, determine the theoretical variogram as illustrated in Fig. 1.

Once we have estimated the theoretical model, we can use it to derive semivariances at all locations and solve the kriging weights. The ordinary kriging (OK) weights are solved multiplying the covariances:

$$\lambda_0 = \mathbf{C}^{-1}.\mathbf{c_0}; \quad C(|h| = 0) = C_0 + C_1 \tag{4}$$

where C is the covariance matrix derived for $N \times N$ observations and c_0 is the vector of covariances at a new location. Note that the C is in fact a $(N+1) \times (N+1)$ matrix if it is used to derive kriging weights, since one extra row and column are used to ensure that the sum of weights is equal to one:

$$\begin{bmatrix} C_{(s_1,s_1)} & \cdots & C_{(s_N,s_1)} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ C_{(s_N,s_1)} & \cdots & C_{(s_N,s_N)} & 1 \\ 1 & \ldots & 1 & 0 \end{bmatrix}^{-1} . \begin{bmatrix} C_{(s_0,s_1)} \\ \vdots \\ C_{(s_0,s_N)} \\ 1 \end{bmatrix} = \begin{bmatrix} w_1(s_0) \\ \vdots \\ w_N(s_0) \\ \varphi \end{bmatrix} \tag{5}$$

where φ is the *Lagrange multiplier*. After calculating the weights, the prediction is then given by Eq. 1.

When the experimental variogram is distinct for two or more directions, we have an anisotropic phenomenon, as can be seen in the ellipse drawn in Fig. 2. The anisotropy is calculated considering a certain angle from 0 to 180°, and a factor given by

$$Factor = \frac{a_2}{a_1} \tag{6}$$

where a_1 and a_2 are the biggest and smallest radius of the ellipse, respectively. This factor varies between 0 and 1, with 1 being an isotropic model. Therefore, in case of anisotropy, five parameters are used to estimate the theoretical variogram model: nugget, sill, range, angle, and the anisotropy factor.

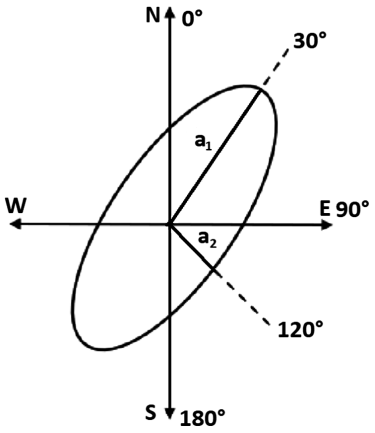

Fig. 2. Example of anisotropy and its parameters.

2.2 KNN

K-Nearest Neighbor (KNN) [14] is a supervised machine learning algorithm used for both classification and regression problems. It is a non-parametric approach that uses training data directly for classification. More specifically, the KNN algorithm classifies a new point based on the training set points that are close to it.

Given a training data set $\{(s_1, q_1), \ldots, (s_N, q_N)\}$, with N points, each point (s, q) consists of a vector $s \in R^L$ and a label $q \in \{1, \ldots Q\}$. Let $s_0 = (p_1, \ldots p_L)$ be a new point not yet classified (i.e. without label). In order to classify this new point, the KNN algorithm calculates the distance between s_0 and the other points in the training set using a measure of similarity. The K nearest points (i.e. with smaller distances) in relation to s_0 are then stored. In the sequel, it is verified which is the most frequent label among the K neighbors, and this elected label q is associated with the new point.

A well-known measure of similarity and also used in this work is the Euclidean Distance, which is defined by

$$d(\mathbf{s}, \hat{\mathbf{s}}) = \sqrt{\sum_{i=1}^{L} (p_i - \hat{p}_i)^2} \tag{7}$$

where p_i and $\hat{p}_i$ are elements of vectors $\mathbf{s}$ and $\hat{\mathbf{s}}$, respectively.

2.3 K-Means

K-means [14] is one of the simplest unsupervised learning algorithms that solve the clustering problem. This clustering algorithm partitions the database into U clusters, where the value of U is provided by the user.

The k-means algorithm starts by initializing a set of U centroids, one for each cluster. A widely used initialization method is the random selection among the points in the database. Each point is then associated with the nearest centroid based on a measure

of similarity (e.g. Euclidean Distance) in order to build the groups. After that, the centroids are recalculated. The "new" centroid is the average of the "old" cluster points. This process is repeated until the centroids are no longer modified.

2.4 Genetic Algorithm

Genetic algorithm [15] is a bio-inspired optimization technique that uses natural evolution theory as inspiration. It is structured by a set of possible solutions. This set is called "population" and each possible solution contain a set of parameters, which we call a "chromosome".

Initially, a population with randomly generated individuals/chromosomes is created. Then, we apply a fitness function to each chromosome to evaluate its goodness or how well it solved the problem in analysis. Then, a selection operator chooses the individuals who will pass to the next generation. Finally, the crossover and mutation operators are applied to the individuals chosen by the selection operator. The algorithm runs until a stopping criterion is fulfilled, such as the number of generations.

We used the R package "ga" [16] when performing tests.

2.5 Differential Evolution

The Differential Evolution algorithm is an optimization technique similar to genetic algorithms in its structure. Similar operators like crossover, mutation and selection are used, however this algorithm focus on the mutation operation to generate the population of individuals/chromossomes. In other words, the DE perturbs the population with of randomly selected chromosomes by using their components to construct trial vectors. The recombination (crossover) operator efficiently shuffles information about successful combinations, enabling the search for a better solution space [17].

We used the R package "DEOptim" [18] when performing tests.

2.6 Particle Swarm Optimization

The PSO uses similar structure from DE and GA algorithms to evaluate each individual (also called particles), however, instead of using crossover, mutation, and selection methods, new solutions are created by "moving" the particle around the search space based on two factors: a) The best fitness achieved by the particle so far b) The best fitness achieved by any particle in the neighborhood [19].

The structure of a particle consists of four parameters, current location (identical to the GA chromosome), current velocity, best local fitness and best global fitness. For every iteration, each particle is updated with a new location based on the other 3 parameters. Nowadays PSO has been successfully applied to various real world problems and it represents current state of art nature inspired algorithm [19].

2.7 Cost Function and Evaluation Metric

The cost function used as the fitness function of all algorithms were the interpolation error cost function [8]. We obtain the error by applying the Kriging at each data point

(Leave-one-out Cross Validation) of the Train database using the parameters of the solution, and calculating the Mean Squared Error after all points have been tested.

To evaluate and compare the algorithms, we applied the 10-fold cross validation, where 10 iterations of tests are performed with 90% train data and 10% test data. Every version of the test database is different from each other. After all iterations, we calculate the mean of the MSE (Mean Squared Error) of each.

3 Methodology

The main objective of this research is to study the behavior and results of the three bio-inspired algorithms, Genetic Algorithm, Differential Evolution, and Particle Swarm Optimization in the optimization of the Kriging parameters: Nugget, Sill, Range, Factor, and Angle.

In order to test the algorithms, we propose the application of Abedini's Cluster Kriging methodology but improving the technique by applying a solution to allocate unknown points to the clusters obtained by the K-means clustering technique. In this scenario, we first apply a preprocessing step using standardization algorithms and treatment of outliers, after that the K-means is used to find k groups of the data, in this clustering step, the KNN method is used to improve the clustering groups. For each k group of data, the bio-inspired algorithms are used to find parameters of the Kriging method. Lastly, the unknown points are allocated to the previously determined clusters using also the KNN method, these points are then interpolated using the kriging parameters and data of that given cluster.

To analyze the results, we applied the 10-fold cross-validation method. In each of the 10 iterations of this method, the data is split into 90% train data, where we apply k-means and the optimization methods (GA, DE, and PSO), and 10% test data (Each iteration has different test data), where we classify using the KNN method into the clusters to apply the kriging method. Summarizing, at each n-fold, we have these steps:

1. Preprocess Data;
2. Partition data into 90% train and 10% test;
3. Split train data into k clusters;
4. Select kriging parameters for each cluster individually via bio-inspired techniques;
5. Allocate each data from test database to a cluster via KNN method;
6. Apply kriging to each data from test database with their respective cluster and parameters;
7. Calculate Mean Squared Error.

The complete flow chart can be seen in the Fig. 3.

3.1 Data Preprocessing

In the data preprocessing step, the spatial information x and y, and the target variable, in this case zinc, were normalized between 0 and 1. This was important to ensure that every variable has the same weight in the clustering process and to avoid cluster overlapping. In the sequel, we used the Z-score test with 99% confidence to remove outliers.

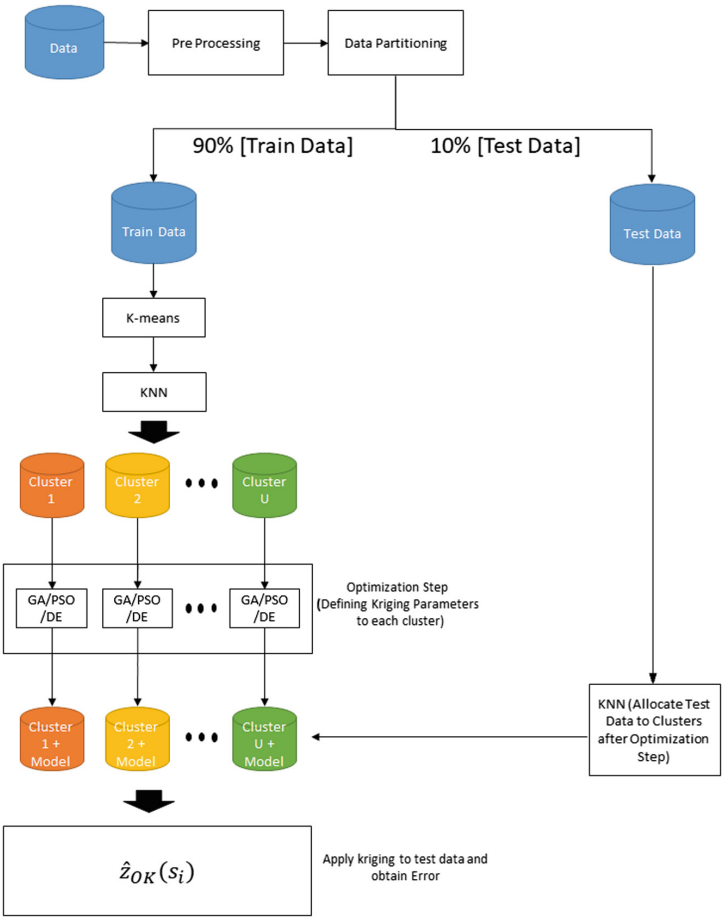

Fig. 3. Flowchart of the proposed method to evaluate the bio-inspired algorithms.

3.2 Data Clustering

After we preprocess the data and partition the data, the train database is split into k clusters using spatial information x and y, and the target variable. As shown in [13], the clustering process often results into overlapped clusters. These overlapping cause problems when allocating unknown (new) data into clusters. To minimize this, all data were previously normalized between 0 and 1, then, each point were classified again into one of the groups found by K-means. This process ensured that all groups were not overlapped. In the following figures we illustrate this scenario, where in Fig. 4(a), especially when the number of clusters are bigger than 4, we do not see a spatial uniformity as observed in Fig. 4(b) where the normalization and KNN was applied. The black circle highlighted in Fig. 4(a), demonstrate an overlapped clusters.



Fig. 4. K-means results (a) Without normalization, (b) With normalization step.

3.3 Optimization Phase

For each cluster obtained by the K-means technique, we apply an optimization technique to find optimal parameters, nugget, sill, range, factor and angle of the Kriging phase. The accuracy of the interpolation are directly correlated to how good these parameters are.

Each bio-inspired algorithm was evaluated based on the best set of parameters found. These parameters are tested when interpolating the test database and the accuracy and error is calculated.

3.4 Classification and Kriging

After we obtain the clusters and their respective kriging parameters, each data of the test database is allocated to a cluster based on KNN method. The KNN method receive the spatial location of the new or unknown point (x and y) and set its cluster based on the nearest neighbors, the most common cluster label found in these neighbors is set to the new point.

In the kriging (interpolation) phase, this new point only takes account the data from its cluster allocated in the previous step, and consequently the parameters found by the optimization techniques for that group.

4 Tests and Results

In this section, we discuss configuration of the tests and show the results.

4.1 Database (Study Area)

The database selected in this research represents the mountainous region of Wolfcamp Aquifer in West Texas/New Mexico. The area has been studied because of its potential site for nuclear-waste disposal. In the wolfcamp aquifer, 85 measurements were collected along this region, each representing piezometric wells. This study area has used

in other works [13, 20] and been classified as an anisotropic data area and irregularly spaced. In Fig. 5, the location of the collected data is shown.

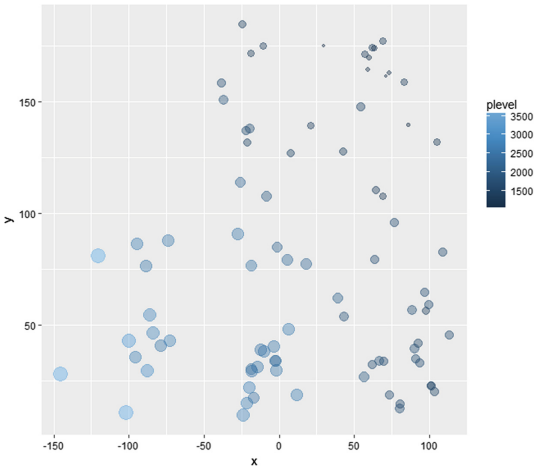

Fig. 5. Wolfcamp aquifer study area, color and size of points indicates the value of target variable. (Color figure online)

4.2 Experiments

In order to evaluate and compare the results for the three bio-inspired algorithms, a manual tuning was performed. The population size and number of iterations values were the same for all algorithms, and other specific parameters of each algorithm were tested manually. The best parameters found for each algorithm are shown in Table 1.

The chromosome (GA and DE) and particle (PSO) had the following configuration:

- Nugget;
- Sill;
- Range;
- Factor;
- Angle;

Some parameters of the experimental variogram were fixed for tests simplicity, such as number of lags, model type and nugget effect with values 10, Exponential and 0 respectively.

The range of each parameter to be optimized were defined as shown in Table 2.

Table 1. Bio-inspired techniques parameters used in tests.

Algorithm	Parameter	Value
GA	Population size	100
	Iterations	10
	Crossover probability	0.8
	Mutation probability	0.1
	Elitism	5% of population
	Selection method	Roulette
	Crossover method	Deep [21] real coded
	Mutation method	Deep [22] real coded
DE	Population size	100
	Iterations	10
	Crossover probability	0.6
	Differential weight	0.2
	Method	de/rand/1/bin
PSO	Particles	100
	Iterations	10
	Social constant	2
	Cognitive constant	2
	Inertia range	0.4 to 0.8

Table 2. Lower and Upper bounds of variables.

Bound	Nugget	Range	Sill	Angle	Factor
Lower bound	0	0	0	0	0
Upper bound	0	Max distance between 2 points	Variance of target variable * 5	180	1

Table 3. MSE results for each bio-inspired algorithm for 2 to 5 clusters (test data).

Algorithm		2 Clusters	3 Clusters	4 Clusters	5 Clusters
GA	Mean	0.091	0.062	0.114	0.138
	Stand. Dev.	0.074	0.066	0.107	0.123
DE	Mean	0.082	0.086	0.099	0.178
	Stand. Dev.	0.063	0.078	0.109	0.209
PSO	Mean	0.093	0.073	0.098	0.132
	Stand. Dev.	0.081	0.059	0.097	0.116

As stated earlier, we used the 10-fold cross validation method. Consequently, 10 tests were ran for each number of clusters for each algorithm. The results can be seen in Table 3.

The best overall result for all algorithms were achieved using 3 clusters with GA. With 5 clusters the algorithms achieved the highest MSE of all configurations. The GA algorithm performed the best with 3 clusters, where the DE algorithm found the best results with 2 clusters, with 4 or 5 clusters the PSO algorithm performed the best.

We can infer that a high number of clusters does not imply a better accuracy of the method, since the worst result is obtained using 5 clusters for all tested algorithms. Therefore, a step for choosing the best number of groups could be interesting for the methodology. Some indicators of clustering goodness, such as Silhouette or Dunn, could be applied to further study this behavior.

Analyzing the boxplot in Fig. 6, we can interpret that the error is sensible to the clustering process, where some occurrences as the outliers found with 3, 4 and 5 clusters have high impact in the Kriging interpolation process.

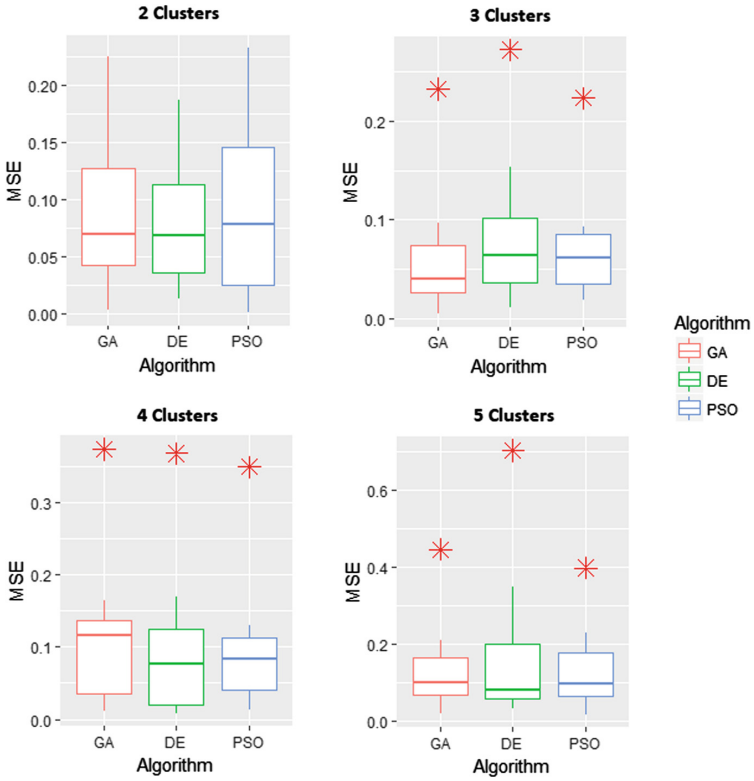

Fig. 6. Boxplot of the obtained results for each algorithm and cluster from 2 to 5.

5 Conclusions

The results obtained with the proposed methodology using data clustering, evolutionary techniques demonstrated a similar behavior in all algorithms (GA, DE and PSO). Some issues and improvements that must be highlighted:

- Computational cost is big with bio-inspired techniques and interpolation cost function when optimizing kriging parameters.
- The algorithms can probably achieve even better results with more generations and bigger population.
- A big standard deviation infer that the kriging process is sensitive to the cluster obtained in the K-means process.

According to our findings, we conclude that the bio-inspired algorithms have similar behavior when estimating parameters for this study case and parameters. Nevertheless, it is obvious that further research is needed. Future works include evaluating other clustering algorithms and evaluating them with some indicators, adding figures of metric and kriging maps to measure the quality of the formed groups. Evaluate others bio-inspired optimization methods is another important issue to be addressed, considering the use of information visualization techniques to analyze the evolutionary behavior and convergence curve.

References

1. Hengl, T.: A Practical Guide to Geostatistical Mapping. 52 edn. (2009)
2. Tugrul, B., Polat, H.: Privacy-preserving kriging interpolation on distributed data. In: Murgante, B., et al. (eds.) ICCSA 2014. LNCS, vol. 8584, pp. 695–708. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09153-2_52
3. Schernthanner, H., Steppan, S., Kuntzsch, C., Borg, E., Asche, H.: Automated web-based geoprocessing of rental prices. In: Gervasi, O., et al. (eds.) ICCSA 2017. LNCS, vol. 10407, pp. 512–524. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62401-3_37
4. Ocal, D., Kentel, E.: A GIS tool to estimate flow at ungaged basins using the map correlation method. In: Gervasi, O., et al. (eds.) ICCSA 2017. LNCS, vol. 10407, pp. 377–391. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-62401-3_28
5. Wei, Z., Liu, Z., Chen, Q.: Ga-based kriging for isoline drawing. In: 2010 International Conference on Environmental Science and Information Application Technology (ESIAT), vol. 2, pp. 170–173. IEEE (2010)
6. Xialin, Z., Zhengping, W., Zhanglin, L., Chonglong, W.: An intelligent improvement on the reliability of ordinary kriging estimates by a GA. In: 2010 Second WRI Global Congress on Intelligent Systems (GCIS), vol. 2, pp. 61–64. IEEE (2010)
7. Gonçalves, I., Kumaira, S., Guadagnin, F.: A machine learning approach to the potential-field method for implicit modeling of geological structures. Comput. Geosci. **103**, 173–182 (2017)
8. Li, Z., Zhang, X., Clarke, K., Liu, G., Zhu, R.: An automatic variogram modeling method with high reliability fitness and estimates. Comput. Geosci. **120**, 48–59 (2018)

9. Abedini, M.J., Nasseri, M., Burn, D.H.: The use of a genetic algorithm-based search strategy in geostatistics: application to a set of anisotropic piezometric head data. Comput. Geosci. **41**, 136–146 (2012)
10. Rat Atalay, F., Ertunç, G.: Metaheuristic kriging: a new spatial estimation method. Hacet. J. Math. Stat. **46**(3), 483–492 (2017)
11. Shang, X., Ma, P., Yang, M.: An improved kriging model based on differential evolution. In: Proceedings of The 9th EUROSIM Congress on Modelling and Simulation, EUROSIM 2016, no. 142, pp. 356–361 (2018)
12. Wang, Z., Chang, Z., Luo, Q., Hua, S., Zhao, H., Kang, Y.: Optimization of riveting parameters using kriging and particle swarm optimization to improve deformation homogeneity in aircraft assembly. Adv. Mech. Eng. **9**(8), 1687814017719003 (2017)
13. Abedini, M.J., Nasseri, M., Ansari, A.: Cluster-based ordinary kriging of piezometric head in West Texas/New Mexico-testing of hypothesis. J. Hydrol. **351**(3–4), 360–367 (2008)
14. Witten, I.H., Frank, E., Hall, M.A., Pal, C.J.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, Burlington (2016)
15. Goldberg, D.E., Holland, J.H.: Genetic algorithms and machine learning. Mach. Learn. **3**(2), 95–99 (1988)
16. Scrucca, L.: GA: a package for genetic algorithms in R. J. Stat. Softw. **53**(4), 1–37 (2013)
17. Karaboğa, D., Ökdem, S.: A simple and global optimization algorithm for engineering problems: differential evolution algorithm. Turk. J. Electr. Eng. Comput. Sci. **12**(1), 53–60 (2004)
18. Mullen, K.M., Ardia, D., Gil, D.L., Windover, D., Cline, J.: DEoptim: an R package for global optimization by differential evolution. J. Stat. Softw. **40**, 1–26 (2009)
19. Mishra, K.K., Tiwari, S., Misra, A.K.: A bio inspired algorithm for solving optimization problems. In: 2011 2nd International Conference on Computer and Communication Technology, pp. 653–659 (2011)
20. Cressie, N.: Fitting variogram models by weighted least squares. J. Int. Assoc. Math. Geol. **17**(5), 563–586 (1985)
21. Deep, K., Thakur, M.: A new crossover operator for real coded genetic algorithms. Appl. Math. Comput. **188**(1), 895–911 (2007)
22. Deep, K., Thakur, M.: A new mutation operator for real coded genetic algorithms. Appl. Math. Comput. **193**(1), 211–230 (2007)