An Optimization-Based Methodology for the Definition of Amplitude Thresholds of the Ground Penetrating Radar

Eslam Mohammed Abdelkader*[1], Mohamed Marzouk[2], and Tarek Zayed[3]

ABSTRACT

Existing infrastructure is aging while the demands are growing for a better infrastructure system in response to the high standards of safety, health, population growth, and environmental protection. Bridges are subjected to severe deterioration agents such as variable traffic loading, deferred maintenance, cycles of freeze and thaw, etc. The development of Bridge Management Systems (BMSs) has become a fundamental imperative nowadays due to the huge variance between the need for maintenance actions, and the available funds to perform such actions. Condition assessment is regarded as one of the most critical and vital components of BMSs. Ground Penetrating Radar (GPR) is one of the non-destructive techniques (NDTs) that are used to evaluate the condition of bridge decks which are subjected to the rebar corrosion. There is a major issue associated with the GPR which is the absence of a scale for the amplitude values. The objective of the proposed model is to compute standardized amplitude thresholds for corrosion maps. The proposed model considers eight un-supervised clustering algorithms to obtain the thresholds. The proposed model incorporates a multi-objective optimization-based methodology that employs three evolutionary optimization algorithms to calculate the optimum thresholds which are: 1) genetic algorithm, 2) particle swarm optimization algorithm, and 3) shuffled frog-leaping algorithm. Five multi-criteria decision- making techniques are used to provide a ranking for the solutions. Finally, group decision-making is performed to aggregate the results and obtain a consensus and compromise solution. The standardized thresholds obtained from the proposed methodology are: -16.7619, -8.8161, and -2.9744 decibels.

Keywords: Bridge Management System, ground Penetrating Radar, non-destructive techniques, corrosion, amplitude thresholds, evolutionary optimization algorithms, multi-criteria decision-making.

[1] *Ph.D. student, Department of Building, Civil, and Environmental Engineering, Concordia University, Montreal, QC, Canada. Corresponding author, E-mail: eslam_ahmed1990@hotmail.com, corresponding author.*

[2] *Professor of Construction Engineering and Management, Structural Engineering Department, Faculty of Engineering, Cairo University, Egypt.*

[3] *Professor, Department of Building, Civil, and Environmental Engineering, Concordia University, Montreal, QC, Canada.*

1 INTRODUCTION

Infrastructure systems refer to systems that support the prevailing of the society. Infrastructure systems are divided into: bridges, highways, dams, waste water systems, sewer water systems, etc. Existing infrastructure is vulnerable to high levels of deterioration. Therefore, billions of dollars should be invested every year in order to maintain the desired levels of standards to the customers. The deterioration in Canada's infrastructure systems is mainly because of two main reasons: 1) the decrease in the investment of the infrastructure systems, and 2) most of the infrastructure systems were constructed relatively a long time ago.

The infrastructure investment in Canada has declined extensively over 40 years from the late of the 1950s to the mid of the 2000s, whereas the infrastructure investment peak was 3% of the Gross Domestic Product (GDP) in the late 1950s and it decreased to 0.4% of the GDP in 1979 (Mackenzie 2013). Most of Canada's infrastructure is relatively old because most of them were constructed between the 1960s and 1970s (Statistics Canada 2014). As per Canada Infrastructure Report Card, 40% of infrastructure systems is in a "Good" condition, 40% of infrastructure systems within 20 years will be in a "Fair" condition, 40% of infrastructure systems within 40 years will be in a "Poor" condition, 40% of infrastructure systems within 60 years will be in a "Very Poor" condition (Felio 2016).

Consequently, the infrastructure deficit will increase significantly where Canada's municipal infrastructure deficit is $123 billion and it is growing by $2 billion per year. The deficit is divided into four main components which are: 1) $31 billion for water and waste water systems, 2) $21.7 billion for transportation and $22.8 billion for transit, 3) $7.7 billion for waste management, and 4) $40.2 billion for community, cultural, and social infrastructure. One-third of infrastructure is either "Fair, Poor or Very Poor (Mirza 2007). Based on the aforementioned

statistics, Canada must invest $66 billion to maintain and repair roads and bridges between 2013 and 2023. Moreover, the transit systems require an annual investment of $4.2 billion to repair and replace existing assets (Mckinsey 2013; Mirza 2007).

Bridges are subjected to aggressive influences such as overloading, chloride ingress, cycles of the freeze and thaw, earthquakes, etc. Thus, they are more likely to deteriorate significantly. The overall condition of the bridges and roads in Canada is "Good" where 57% of the bridges are in "Good" condition, and 22% of the bridges are in "Fair" condition (Felio 2016). The number of highway bridges in Canada is 75,000 where their average age is 24.5 years in 2007 compared to a mean service life of 43.3 years (Statistics Canada 2009a). This means that the bridges in Canada have passed 57% of their useful lifetime (Statistics Canada 2009a). Bridges in Quebec have the highest average age of 31 years followed by Nova Scotia with an average age of 28.6 years (Statistics Canada 2009b).

Bridge Management Systems (BMSs) have become a necessity nowadays in order to provide a tool for the government agencies to manage a large network of bridges under some constraints such as limited resources (budget). AASHTO defined Bridge Management System (BMS) as "a system designed to optimize the use of available resources for inspection, maintenance, rehabilitation and replacement of bridges" (AASHTO Highways subcommittee on Bridges and Structures 2011). There are five main components of BMS which are (Czepiel 1995): 1) database for data storage, 2) condition rating model, 3) deterioration model, 4) cost model, and 5) optimization model for running the system.

Condition assessment is considered as one of the main fundamental and vital pillars of BMS. Visual inspection is considered as one of the most common techniques in the condition

assessment of bridge decks. However, it is a very subjective technique and deals with only the defects visible on the surface. Non-destructive techniques (NDTs) have gained positive recognition in the condition assessment of concrete structures. Ground Penetrating Radar (GPR) is one of the NDTs that are used to evaluate corrosion of reinforcing rebars in concrete bridge decks based on the transmitted electromagnetic waves from the antenna. Condition assessment plays a very role in the prioritization plan for the maintenance, repair, and rehabilitation (MR&R) actions. Developing an accurate corrosion index is one of the pillars of the condition assessment. Thus, the presence of standardized thresholds for the GPR is very essential in order to provide an equal basis for comparison between the different concrete bridge decks.

The proposed model utilizes Ground Penetrating Radar in order to overcome the drawbacks and the flaws of the visual inspection. Nevertheless, GPR suffers from some limitations where one of the main limitations that there are no clear or exact thresholds for the GPR scale. The objectives of the current research are as follows:

1. Review the previously-developed and identify their shortcomings.
2. Compute the standardized amplitude thresholds for the GPR
3. Develop a corrosion map and calculate a corrosion index for the bridge deck.

2 LITERATURE REVIEW

Few contributions have been made in the field of defining thresholds for the GPR amplitudes where there is no clear value for the thresholds that define the different categories of corrosion. Dinh and Zayed (2016) calculated a bridge deck corrosiveness index (BDCI) based on the fuzzy set theory. They developed a corrosion map for concrete bridge decks. The thresholds of the corrosion map were calculated based on K-means clustering. On the other hand, the model has some limitations because it is based on K-means clustering. K-means clustering is an un-

supervised learning technique which means that it is highly dependent on the input dataset since there is no prior information about the model. Thus, if another bridge deck is investigated, the amplitude thresholds will be different consequently.

Shami (2015) calculated the thresholds based on some goodness of fit tests such as Kolmogorov-Smirnov test, Anderson Darling, and chi-squared test. These tests were used to select the most feasible probabilistic distribution that fits the dataset. They concluded that the threshold that separates the “very severe” category from the “severe” category, and the “Severe” category from the “medium” category follow the logistic distribution. Moreover, they concluded that the threshold that separates the “medium” category from the “good” category follows the triangular distribution. The value of the threshold is calculated based on the median of the probabilistic distribution. The model had some limitations where the most feasible probabilistic distribution was selected based on only 34 bridge decks. However, Fornell (1983) illustrated that the minimum sample size required to perform goodness of fit tests is 200. In addition to that, Grant et al. (2017) stated that the Kolmogorov-Smirnov test is performed when the sample size is more than 50 observations.

Martino et al. (2016) calculated a threshold for the GPR based on the receiver operating characteristic curves (ROC) using half-cell potential as a ground truth. They concluded that -1.6 DB is the universal threshold that separates the healthy areas from the corroded areas in concrete bridge decks. However, the developed methodology can be only used for the binary classification since it is based on the ROC curves, i.e., for the separation of the corroded areas from the non-corroded areas. The developed methodology will not be feasible to provide the amplitude thresholds in the case of the existence of more than two condition categories.

Based on the previous limitations, the research presents a model that is capable of calculating standardized thresholds of the GPR and for any number of bridge condition categories. The proposed methodology is a hybrid model that is divided into three main modules which are: 1) clustering module. 2) multi-objective optimization module, and 3) decision-making module. The clustering module utilizes eight clustering algorithms for a different number of bridges. A multi-objective optimization module is designed to calculate the standardized thresholds based on the output from the clustering module, and based on four objective functions. The multi-objective optimization module integrates three evolutionary algorithms which are: 1) genetic algorithm, 2) particle swarm optimization algorithm, and 3) shuffled frog-leaping algorithm. The decision-making is utilized in order to select the most feasible solution among the optimum solutions.

3 MODEL DEVELOPMENT

The framework of the proposed methodology is shown in Figure 1. The proposed methodology is a hybrid model that integrates evolutionary algorithms, un-supervised clustering algorithms, and multi-criteria decision-making techniques. The multi-objective optimization module utilizes three evolutionary algorithms which are: 1) genetic algorithm, 2) particle swarm optimization algorithm, and 3) shuffled frog-leaping algorithm. These three algorithms are selected due to their capability of solving the discrete and continuous optimization problems efficiently. The first step is to survey the bridge decks using the ground penetrating radar. Ground penetrating radar is one of the non-destructive techniques that are used for field investigation in structural engineering. Ground Penetrating radar can determine the subsurface structure easily and accurately. Moreover, it has the capability of locating metallic and non-metallic objects. GPR transmits pulsed electromagnetic waves from the transmitting antenna which is located on the ground surface and signals are then received by the receiving antenna.

The proposed model utilizes GPR in order to evaluate the corrosion of the reinforcement rebars in the concrete bridge decks. GPR system is composed of data collection system and antennas. There are two types of antennas which are: 1) mono-static antenna, and 2) bi-static antenna. Mono-static antennas are composed of one antenna that performs both transmitting and receiving. Bi-static antennas include separate antennas for transmitting pulses and receiving those that are reflected. There are three basic components of GPR system which are: 1) display unit, 2) control unit, 3) an antenna, and 4) cart. The display unit is used to display the recorded data such as laptop. Control unit controls the operation of transmitting and receiving electromagnetic pulses. The antenna is used to perform the task of transmitting electromagnetic waves and receive the reflected pulse.

Then, the scanned profiles are imported into the GSSI RADAN7 software in order to extract the needed information. GSSI RADAN7 software is used to extract the amplitude values of the top reinforcing rebars. The numerical-amplitude method is used to interpret the corrosion of the bridge decks. Numerical amplitude method depends on the value of the amplitude of the reflected waves from the top layer of reinforcement. The higher the amplitude the better the condition of the bars will be. On the other hand, the lower the amplitude the higher the corrosion the reinforcement bars will be and consequently, the lower condition state the bridge deck will be. This method main drawback is its lack of a clear value for the thresholds that define the different categories of corrosion. For example, the profiles of one bridge deck may have amplitude values from 10 dB to -5 dB, where 10 dB represents the best condition and -5 dB represents the worst for that bridge. Meanwhile, another of Bridges' profiles may have amplitude values that range from -5 dB to -40 dB, where -5 dB represents the best condition and -40 dB represents the worst condition. A Microsoft Excel spreadsheet is generated containing some

important parameters retrieved from the GPR profiles such as scan number, two-way travel time, and normalized amplitude for each reinforcement rebar.

Then, the depth correction is performed based on the methodology developed by Barnes et al. (2008). The main objective of this step is to remove the effect of the depth on the target data because there is an attenuation of the electromagnetic waves associated with the deeper targets. Depth-corrected amplitudes provide a more accurate assessment of the amplitude values of the reinforcement rebars. The depth correction was performed as follows: 1) the data points were divided into time bins; e.g. 0.5 ns, 2) 90^{th} percentile point for each time bin is calculated assuming that the chloride content is consistent for the 90^{th} percentile of the normalized amplitude at each time bin, i.e., amplitude values above the 90^{th} percentile are not affected by deterioration, 3) regression is applied to fit the 90^{th} percentile points, and 4) correcting the data by forcing the 90^{th} percentile to be zero decibel (dB). After the depth correction, the threshold values should be calculated in order to delineate the areas of corrosion.

The third step of the proposed methodology is the clustering module where a group of bridge decks is used as an input for the proposed methodology. Several clustering algorithms are applied because each one of them depends on a certain calculation methodology which generates different clusters, and consequently different thresholds. Eight clustering algorithms are utilized which are: K-means, fast K-means, kernel K-means, K-medoids, expectation maximization, fuzzy C-means, X-means, and agglomerative clustering. The multi-objective optimization module takes into consideration any number of bridge decks (based on the available dataset) and it calculates the standardized thresholds based on four objective functions. The first three objective functions tend to find the optimum threshold based on a local search, i.e., dealing with each threshold individually. On the other hand, the fourth objective function tends to find the

optimum threshold based on a global search. The multi-objective optimization module incorporates three evolutionary algorithms which are: genetic algorithm, particle swarm algorithm, and shuffled frog-algorithm. The output of this module is the combined Pareto frontier points obtained from the three algorithms where each one of the candidate solutions is represented in a three-dimensional space, i.e., threshold (1), threshold (2), and threshold (3).

Decision-making module is used to calculate the most feasible solution among the optimum solutions obtained from the multi-objective optimization module. Five multi-criteria decision-making techniques are implemented which are: 1) WSM (Weighted Sum Model), 2) COPRAS (Complex Proportional Assessment), 3) VIKOR, 4) GRA (Grey Relational Analysis) and 5) TOPSIS (Technique for Order Preference by Similarity to Ideal Solution). Each one of the decision-making techniques provides a distinct ranking for the alternatives. Group decision-making is performed to aggregate the ranking of the alternatives into one final ranking for the alternatives, i.e., obtain the best compromise solution. The alternative with the first ranking represents the standardized thresholds of the GPR. After calculating the thresholds, a corrosion map can be developed for any bridge deck. Surfer 12 is a plotting and mapping software that is utilized to develop the corrosion map for the concrete bridge decks. Finally, a corrosion index (CI) can be calculated using the following Equation.

$$CI = \frac{\sum_{c=1}^{4} Q_c \times W_c}{\sum_{c=1}^{4} Q_c} \quad (1)$$

, where

Q_c represents the quantity (area) of a bridge element in condition category c. W_c represents the weighting factors for a bridge element in condition category c. The weighting factors for the

"good", "medium", "severe", and "very severe" condition categories are assumed 100%, 70%, 50%, and 20%, respectively.

INSERT FIGURE 1

3.1 Clustering Module

Clustering is the process of partitioning the dataset into a homogenous set of clusters without having any prior information about the clusters where the points within the same cluster share similar features. The selected clustering algorithms incorporate a combination of soft and hard clustering techniques. Hard clustering is the process of the assignment of data points to only one cluster such as K-means clustering algorithm. On the other hand, soft clustering is the process of the assignment of data points to the clusters with different membership degrees such as fuzzy C-means clustering algorithm (FCM). RapidMiner 7.5 and KNIME 3.3.1 softwares are used as platforms to perform the clustering algorithm. K-means and fuzzy C-means clustering algorithms are only discussed in the following lines due to the limited space.

3.1.1 K-Means Clustering Algorithm

K-means clustering algorithm is based on minimizing the distance between the average squared Euclidean distance between the data points and the clusters' centroids. The main distinct feature between K-means and k-medoids clustering algorithms is that one of the data points represents the centroid of the cluster in the case of k-medoids. K-means algorithm utilizes the mean of the data points. The steps of K-means clustering algorithm are as follows (Sawant, 2015):

1- Select the number of desired clusters K.

2- Select K starting points randomly to be used as initial candidates for clusters' centroids.

3- Calculate the distance between data points and cluster centroids.

4- Assign the data point to the cluster centroid which has the minimum distance between the data point and cluster centroids. The distance is simply the Euclidean distance.

$$d(x_i, C_j) = \sqrt{\sum_{d=1}^{n} (x_{id} - c_{id_i})^2} \quad (2)$$

5- Re-compute the new cluster centroids (centroid is the mean point of the cluster).

6- Repeat steps 3, 4, and 5 until convergence (centroid and data points no longer move).

3.1.2 Fuzzy C-means Clustering Algorithm

FCM is an iterative clustering algorithm where each data point is assigned to one cluster or more based on the membership degrees. FCM was developed by Dunn in 1973 and improved by Bezdek in 1981. FCM is based on minimizing the following objective function (Keskin, 2015).

$$J_w = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left|\left| \left(X_i - C_j\right)^2 \right|\right| \quad (3)$$

, where

m is a fuzzifier constant that is greater than one. u_{ij} denotes the degree of membership of the X_i in the cluster j and it is between zero and one. X_i is a $i - th$ data point in a d-dimensional space. C_j represents the centroid of the $j - th$ cluster. $||*||$ is a norm distance that represents the similarity between the data point and the centroid of the cluster.

FCM starts by randomly initiating the cluster centroid. The second step is to construct the membership matrix. A membership matrix ($U_{(N\times C)}$) is composed of a group of membership degrees. The degree of membership (u_{ij}) can be calculated using Equation (4). The cluster centroids are then updated and can be calculated using Equation (5). The cluster centroids and the membership degrees are iteratively updated until the convergence criteria are satisfied. The convergence criteria is shown in Equation (6). The de-fuzzification process is performed using

Equation (7) where the data point is assigned to the cluster that has the maximum degree of membership.

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\left\| (X_i - C_j) \right\|}{\left\| (X_i - C_k) \right\|} \right)^{\frac{2}{m-1}}} \quad (4)$$

$$C_j = \frac{\sum_{i=1}^{N} u_{ij}^{m} \times X_i}{\sum_{i=1}^{N} u_{ij}^{m}} \quad (5)$$

$$\max_{ij} \{ |u_{ij}^{it+1} - u_{ij}^{it}| \} < \zeta \quad (6)$$

$$D_j = \arg_i \{ max(u_{ij}) \} \quad (7)$$

$$\sum_{j=1}^{C} u_{ij} = 1 \quad (8)$$

, where

D_j represents the de-fuzzified value, which is calculated based on the maximum degree of membership principle. ζ is the termination constant between zero and one. it refers to the number of iteration steps.

3.2 Clustering Validity Approaches

Clustering is an un-supervised algorithm. Therefore, evaluating the output of the clustering algorithms is a matter of great importance. Assessing the clustering algorithms is much more difficult than the supervised algorithms because there is no “ground truth”, i.e., there are no-predefined classes for the domain problem. Moreover, it is very difficult to find the appropriate metrics to evaluate the quality of the generated clusters (Sahani and Bhuyan 2014).

The proposed model utilizes two clustering validity approaches to assess the quality of the generated clusters and to identify the optimal partition of clusters which are: 1) Davies-Bouldin index, and 2) Dunn index. The objective of the clustering validity approaches is to select the most feasible thresholds that ensure that the clusters are compact and well-separated, i.e., maximize the inter-cluster distance (distance between the clusters), and minimize the intra-cluster distance (distance between data points within the same cluster) (see Figure 2).

INSERT FIGURE 2

3.2.1 Davies-Bouldin Index

Davies-Bouldin index is a ratio between the sum of intra-cluster scatter to the inter-cluster separation. The Davies-Bouldin index can be calculated using the following Equation (Davies and Bouldin 1979).

$$DBI = (\frac{1}{k}\sum_{w,v=1}^{k} \max_{v \neq w} \left(\frac{D_w + D_v}{d(c_w, c_v)} \right) \quad (9)$$

, where

D represents the intra-cluster distance. d represents the inter-cluster distance.

The intra-cluster distance (D), and the inter-cluster distance (d) can be calculated using Equations (10), and (11), respectively based on the Euclidean distance principle. The intra-cluster distance is the average distance between the data points and the cluster centroid. The inter-cluster distance is the distance between the centroid of the two clusters.

$$D = \left(\frac{\sum_{h=1}^{t} ||X_a - C_w||}{N_w} \right) \quad (10)$$

$$d = dist(C_w, C_v) = \sqrt{(C_{w1} - C_{v1})^2 + (C_{w2} - C_{v2})^2 \dots\dots\dots\dots\dots \left(C_{wq} - C_{vq}\right)^2} \quad (11)$$

, where

X_a is an arbitrary data point that belongs to a cluster w. C_w and C_v represent the centroid of clusters w and v, respectively. N_w represents number of data points in the cluster w. The smaller Davies-Bouldin index denotes that the clusters are compact, and the centers of the clusters are far away from each other (Sahani and Bhuyan 2014). t is the number of the data points in the cluster. q is the number of the dimensions of the model.

3.2.2 Dunn Index

The Dunn index is used to assess the quality of the clusters and it can be calculated using Equation (12) (Dunn, 1974).

$$DUI = \left(\min_{1 \le v < K,} \left(\min_{1 \le w \le K, v \ne w} \left(\frac{d(x_w, x_v)}{\max_{1 \le k \le K} (D(x_k))} \right) \right) \right) \quad (12)$$

, where

D, and d are defined as above. The larger the Dunn index indicates that the clusters are compact and well-separated (Sahani and Bhuyan 2014).

3.3 Multi-Objective Optimization Module

3.3.1 Formulation of the Optimization Problem

The multi-objective optimization module is used to determine the optimum solutions based on three evolutionary algorithms which are: 1) genetic algorithm, 2) particle swarm optimization algorithm, and 3) shuffled frog- leaping algorithm. Evolutionary algorithms are very effective in solving very complex problems where hill-climbing derivative based algorithms are trapped in local search solutions. Moreover, evolutionary algorithms can handle multiple conflicting objectives directly and simultaneously (Caldas and Norford 2002).

The proposed model assumes that there are four categories for the condition of the bridge deck (very poor, poor, medium and good), i.e., three separating amplitude thresholds. The three separating amplitude thresholds are: threshold (1), threshold (2) and threshold (3), which are denoted as X^-, Y^- and Z^- in the following multi-objective optimization problem. The multi-objective optimization module constitutes four objective functions. The first objective function is to minimize the root mean square error ($RMSE$) of standardized amplitude threshold (1), i.e., minimize the distance between the first threshold obtained from the different clustering algorithms of some bridges, and standardized amplitude threshold (1). The second objective function is to minimize the $RMSE$ of standardized amplitude threshold (2), i.e., minimize the distance between the second threshold obtained from the different clustering algorithms of some bridges, and standardized amplitude threshold (2). The third objective function is to minimize the $RMSE$ of standardized amplitude threshold (3), i.e., minimize the distance between the third threshold obtained from the different clustering algorithms of some bridges, and standardized amplitude threshold (3). The fourth objective is to evaluate the quality of evaluation clusters and it is calculated as the average of the Davies-Bouldin index and Dunn index. The decision variables of the multi-objective optimization problem are the three standardized amplitude thresholds (X^-, Y^- and Z^-).

$$F1 = \min RMSE_1 = \min \sqrt{\frac{\left(\sum_{j=1}^{m}\sum_{i=1}^{n}\left(X_{ij} - n \times m \times X^-\right)^2\right)}{n \times m}} \tag{13}$$

$$F2 = \min RMSE_2 = \min \sqrt{\frac{\left(\sum_{j=1}^{m}\sum_{i=1}^{n}\left(Y_{ij} - n \times m \times Y^-\right)^2\right)}{n \times m}} \tag{14}$$

$$F3 = \min RMSE_3 = \min \sqrt{\frac{\left(\sum_{j=1}^{m}\sum_{i=1}^{n}\left(Z_{ij} - n \times m \times Z^-\right)^2\right)}{n \times m}} \tag{15}$$

$$F4 = \min CLU = \frac{\sum_{i=1}^{n}(DBI - DUI)}{2 \times n} \quad (16)$$

, where

m indicates the number of clustering algorithms. n indicates the number of bridges. X^-, Y^-, and Z^- represent the standardized amplitude threshold (1), standardized amplitude threshold (2), and standardized amplitude threshold (3), respectively. Threshold (1) represents the threshold that separates the "very severe" condition category from the "severe" condition category. Threshold (2) represents the threshold that separates the "severe" condition category from the "medium" condition category. Threshold (3) represents the threshold that separates the "medium" condition category from the "good" condition category. X_{ij} indicates the amplitude threshold (1) obtained from applying the clustering algorithm i for bridge deck j. Y_{ij} indicates the amplitude threshold (2) obtained from applying the clustering algorithm i for bridge deck j. Z_{ij} indicates the amplitude threshold (3) obtained from applying the clustering algorithm i for bridge deck j. CLU represents the clustering index. DBI represents the Davies-Bouldin index.

After defining the optimal solutions from the three evolutionary algorithms, a code is written in Matlab in order to calculate the Pareto frontier points. i.e., set of the Pareto optimal solutions. Consequently, the output of this module is combined sets of Pareto optimal solutions obtained from the three evolutionary algorithms.

3.3.1.1 Genetic Algorithm

Genetic algorithm (GA) is one of the evolutionary algorithms. Genetic algorithm is an optimization tool developed by John Holland in 1975 (Kühn et al. 2013). Genetic algorithm is based on two main processes. The first process is the selection of individuals for the production of the next generation (Garg and Mittal, 2014). The second process is the manipulation of the

selected individual to form the next generation by crossover and mutation (Garg and Mittal 2014). The selection methodology determines which chromosomes are chosen for reproduction and how many off springs are produced. The better individual has a higher chance of being a parent (Garg and Mittal 2014). The flowchart of the genetic algorithm is shown in Figure 3.

The first step is to form a random population of solutions where the solutions are represented in the form of a string called "chromosomes". Each stage a new population of individuals is created and it is called “generation” (Caldas and Norford 2002). Chromosomes consist of genes that carry the set of values for the optimization variables (Elbeltagi et al. 2005). The second step is to calculate the fitness function for each chromosome in the population. The fitness function is used to assess the different chromosomes.

The third step is the selection of the chromosomes. The selection process determines which chromosomes will mate to form the new chromosomes. There are different types of the chromosomes selection strategies which are: roulette wheel selection, rank selection, steady-state selection, elitism, Boltzmann selection and tournament selection. The fourth step is to perform the crossover in order to generate an offspring between the two chromosomes or individuals. There are different types of crossover such as single point crossover, two-point crossover, and uniform crossover. The most common type of crossover is the single-point crossover where a random point is selected at which the remaining genes from one parent to another are swapped (Heidari and Movaghar 2011).

The fifth step is to perform the mutation. The mutation gene is chosen randomly. The process of the mutation occurs by looping through all the genes of the individuals and if a gene is selected for mutation, the gene will be changed by a small value or it will be replaced by a new value Heidari and Movaghar 2011). Mutation is performed in order to ensure the genetic diversity

within the population (Heidari and Movaghar 2011), and to avoid the stagnation around local minima. Finally, a population is generated in each generation and the above processes continue for a certain number of iterations. The chromosomes in the final iteration are the optimal solutions. The genetic algorithm is performed using Matlab optimization Toolbox.

INSERT FIGURE 3

3.3.1.2 Particle Swarm Optimization Algorithm

Particle swarm optimization (PSO) algorithm is a population-based heuristic search algorithm inspired by the social behavior of birds flocking to the desired place in a multi-dimensional space. PSO belongs to the family of "swarm intelligence" algorithms in solving optimization problems. PSO was originally developed by Eberhart and Kennedy in 1995. The PSO starts by creating a population called "swarm" which is composed of individuals called "particles" where each particle adjusts its own flying based on its own flying experience and its companions experience. Each particle represents a candidate solution in a multi-dimensional space where the status of the particle is characterized by its position and velocity and they are updated within each iteration (Zhang and Li 2010).

PSO shares some similarities with the genetic algorithm where each one of them is initialized a population of candidate solutions which are called "chromosomes" in the case of the genetic algorithm, and "particles" in the case of PSO, and they both try to search for the optimum solution within a number of iterations. The main difference between them is that the generation of candidate solutions is done through operators such as mutation and crossover in the genetic algorithm. On the contrary, the natural evolution is done through the exchange of the information between particles. Each particle in the search space moves in the multi-dimensional search space towards its own best flying experience (personal best "$pbest$"), and towards the best experience

so far in the entire swarm (global best "$gbest$") (Kumar and Reddy 2007). PSO is discussed in detail in the following lines.

During the optimization process, the velocity and position of the particle are updated. The velocity and the position of the particles are updated using Equations (17), and (18), respectively (Baltar and Fontane 2008; Yang 2007).

$$v_i(t+1) = w \times v_i(t) + c_1 \times r_1 \times (pbest_i(t) - x_i(t)) + c_2 \times r_2 \times (gbest_i(t) - x_i(t)) \quad (17)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (18)$$

, where

$x_i(t+1)$ represents the updated position vector of the particle i. $x_i(t)$ represents the current position vector of the particle i. $v_i(t+1)$ indicates the updated velocity of the particle i. $v_i(t)$ indicates the current velocity of the particle i. r_1, and r_2 are two uniformly distributed random numbers in the interval [0, 1] where they allow the potential of finding better solutions along the direction which is guided towards the global best, and the personal best. c_1, and c_2 are two constants and they refer to the cognitive learning, and social parameters and they control the effect of personal and global guides. Normally, c_1, and c_2 are assumed 2. w refers to the inertia weight which is used to control the balance between the global and the local experience. A typical range of the inertia weight is between 0.3, and 0.7. It is recommended to start with a large inertia weight at the beginning of the optimization process, and it decreases within the iterations using a damping factor to facilitate the global exploration of the search space. A code is written in Matlab in order to perform the multi-objective particle swarm optimization using Matlab R2013a.

3.3.1.3 Shuffled Frog-Leaping Algorithm

Shuffled frog-leaping (SFL) algorithm is one of the "swarm intelligence" algorithms that are based on simulating the social behavior of frogs when searching for the location that has the maximum amount of the available food. SFL is one of the latest meta-heuristic algorithms that was presented by Eusuff and Lansey in 2003. SFL starts by generating a set of frogs as candidate solutions where the number of dimensions for all frogs is equal to the number of decision variables. In addition to that, an objective function is calculated for each frog (Orouji et al. 2016). SFL combines the strengths of both the genetic-based memetic algorithm (MA), and the social behaviour-based particle swarm optimization algorithm (Wang and Fang 2011).

The frogs are divided into subsets called "memeplexes". The different memeplexes represent different cultures of the frogs where each frog performs a local search within each memeplex. Within each memeplex, the behavior of the frog is affected by the behavior of other frogs in the memeplex where frogs evolve through a memetic evolution process (see Figure 4). Ideas are passed between the memeplexes in a shuffling process after a number of memetic processes where the local search and the shuffling process continue until the convergence criteria are satisfied.

INSERT FIGURE 4

The flowchart of the SFL is depicted in Figure 5. An initial population of "F" frogs is generated in a S-dimensional search space and the dimension of the problem equals to the number of the decision variables (Elbeltagi et al., 2005). A frog i is represented as $X_i = \{X_{i1}, X_{i2}, \dots\dots\dots X_{is}\}$. The entire population is divided into "M" groups of "memeplexes" where each memeplex consists of "N" frogs i.e., $F = M \times N$. The frogs are arranged in a descending order based on the fitness function where the first frog goes to the first memeplex, the second frog goes to the

second memeplex, the M^{th} frog goes to the M^{th} memeplex, and the $M+1$ frog goes to the first memeplex and so on.

The positions with the best and the worst fitness functions are identified which are: X_b, and X_w, respectively. The frog with the global best fitness function is denoted as X_g. Then, a process (frog leaping rule) is applied to improve the position of the frog of the lowest fitness function only in each memeplex similar to the PSO algorithm as shown in Equations (19), and (20).

$$D_i = Rand \times (X_b - X_w) \tag{19}$$

$$X_{new} = X_w + D_i, \text{-}D_{max} <= D_i <= D_{max} \tag{20}$$

, where

$Rand$ represents a random number between 0 and 1. D_i represents the change in the frog's position. D_{max} represents the maximum allowable change in the frog's position. X_w represents the current position of the frog. X_{new} denotes the new updated position of the frog. I represents the number of iterations within each memeplex. If the frog leaping rule generates a better solution, consequently the better solution replaces the worst frog. Otherwise, Equations (19) and (20) are repeated to produce a better solution but the global best solution replaces the best solution in the memeplex. If no improvement can be achieved, a new solution replaces the worst solution randomly.

The SFL performs the independent local search within each memeplex for a predefined number of memetic evolutionary steps. After a predefined number of iterations, global shuffling occurs which allow the frogs among the memeplexes to exchange the information (Venkatesan and Sanavullah 2013). The population is then ranked in a descending order, and the frog of the global best fitness is updated and the process continues until the convergence criteria are satisfied, i.e.,

until reaching maximum number of iterations. A code is written in Matlab in order to perform the multi-objective shuffled frog-leaping optimization using Matlab R2013a.

INSERT FIGURE 5

3.3.1.4 Hypervolume Indicator

There are several metrics that can be used to compare between the multi-objective optimization algorithms such as general distance, inverted generational distance, epsilon indicator, and hypervolume indicator. Hypervolume indicator is the most common performance metric and it is also known as S-metric, hyper-area or Lebesgue measure (Riquelme et al., 2015). The hypervolume indicator is an effective unary approach that can be used to evaluate the performance of the multi-objective optimization algorithms by measuring the quality of the Pareto optimal fronts. An illustration of the hypervolume is shown in Figure 6. The hypervolume indicator calculates the size of the region of the objective space that is covered by a set of Pareto optimal solutions (non-dominated or non-inferior solutions). The hypervolume indicator is the only performance metric that is capable of considering three main aspects which are: diversity, accuracy and cardinality (Riquelme et al., 2015). The hypervolume indicator can be calculated using the following Equation (Nebro et al., 2013). Cardinality refers to the number of solutions exist in the Pareto front, whereas a large number of solutions is more preferred. Accuracy refers to how close the generated solutions to the true Pareto front. Diversity refers to the relative distances, coverage and spread of the generated optimum solutions in the search space.

$$I_{HV} = Volume\left(\bigcup_{s=1}^{|Q|} v_s\right) \tag{21}$$

, where

I_{HV} denotes the hypervolume indicator. v_s represents the hypercube of non-dominated solution s. Q represents the set of the Pareto frontier.

INSERT FIGURE 6

A reference point needs to be identified in order to calculate the hypervolume indicator. The reference point is the point associated with the worst objective function values (nadir point) (Pourbahman and Hamzeh, 2013). A higher Hypervolume indicator indicates that there is a large distance between the Pareto frontier and the reference point (worst solution). Accordingly, the evolutionary algorithm of the highest hypervolume indicator is the optimization algorithm with the best Pareto frontier, i.e., more space is dominated and a better quality of the solutions. The hypervolume indicator describes the closeness of the solutions to the Pareto optimal set. Moreover, it measures the spread of the solutions over the objective space (Fu et al., 2012).

3.3.1.5 Inverted Generational Distance

Inverted generational distance (IGD) is a quality indicator that can be used to compare between the evolutionary algorithms in terms of diversity and convergence where this measure was introduced by Czyzak and Jaszkiewicz in 1998 (Ishibuchi et al, 2014). The IGD utilizes the true Pareto front as a reference point, whereas it measures the Euclidean distance between the elements of the true Pareto front and the approximate Pareto front (Mashwani and Salhi, 2016). The smaller the inverted generational distance, the performance of the optimization algorithm is better in terms of convergence to the Pareto optimal solutions as well as the diversity of the generated solutions. An illustration of the inverted generational distance is depicted in Figure 7. The inverted generational distance can be calculated using Equation (22).

INSERT FIGURE 7

$$IGD = \frac{\sum_{v \epsilon p^*} d(v, A)}{|P^*|} \quad (22)$$

, where

IGD represents the inverted generational distance whereas it can be defined as the average distance from P^* to A. $d(v, A)$ indicates the minimum Euclidean distance between v and the points of A. P^*denotes a set of uniformly distributed points along the true Pareto front. A represents set of non-dominated solutions obtained by the optimization algorithm (an approximate of the Pareto front). In order to obtain a low IGD, A must be close to P^*. An equal population size of 250 is used in order to provide equal basis of comparison between the three evolutionary algorithms as well as same number of iterations.

3.4 Decision-Making Module

The objective of the decision making is to calculate the standardized thresholds of the GPR based on the output from the multi-objective optimization module, i.e., to select the best solution among the set of the Pareto optimal solutions with respect to the relevant attributes. The selection of a solution among the set of finite Pareto optimal solutions is usually called a “posterior” approach. The Pareto front consists of numerous optimal solutions, which are preferred based on the decision-making requirements. Usually, multi-criteria decision-making is employed to select the best solution among the set of Pareto optimal solutions. The hybrid optimization-decision making model is shown in Figure 8. The proposed model considers a group of multi-criteria decision-making techniques that are different in nature in order to obtain a more reliable and comprehensive solution.

INSERT FIGURE 8

3.4.1 Multi-Criteria Decision-Making

Multi-criteria decision-making methods are a group of methods that allow aggregation and consideration of different attributes in order to rank alternatives and choose the best alternative (Mulliner et al. 2013). The proposed model performs multi-criteria decision-making techniques to select the most feasible solution based on the results obtained from the multi-objective optimization module. Shannon entropy is employed to compute the weights of the attributes and five different decision-making techniques are used in this research to rank the alternatives. The proposed model utilizes five multi-criteria decision-making techniques which are: 1) WSM, 2) COPRAS, 3) VIKOR, 4) GRA, and 5) TOPSIS.

3.4.1.1 Shannon Entropy

As mentioned before, the optimum solutions are obtained based on a multi-objective optimization problem. The four objective functions are treated as the attributes in the decision-making module. The weights of the attributes are calculated based on the Shannon entropy method where the weights of the attributes are calculated based on the degree of index dispersion (Akyene et al. 2012). Shannon entropy is based on information theory where it assigns a smaller weight to the attribute if this attribute has similar values across the alternatives, i.e., if the measures of performance of the alternatives of a given attribute are relatively equal, therefore this attribute is considered as relatively unimportant by the decision maker.

The first step is to calculate the Weight (P_{ij}) which is calculated using Equation (23).

$$P_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}} \quad (1 \leq i \leq m, 1 \leq j \leq n) \tag{23}$$

, where

P_{ij} represents the weight of the $i-th$ alternative with respect to $j-th$ attribute. x_{ij} represents the measure of performance of the $i-th$ alternative with respect to $j-th$ attribute. The terms m and n indicate the number of alternatives and number of attributes, respectively.

The second step is to calculate the Entropy value and it is calculated using Equation (24).

$$e_j = -k * \sum_{i=1}^{m} P_{ij} * lnP_{ij} \ (1 \le i \le m, 1 \le j \le n) \quad (24)$$

, where

$$k = \frac{1}{ln\,(m)}$$

, where

e_j refers to the Entropy value of the $j-th$ attribute.

The third step is to calculate the variation coefficient for different attributes and it is calculated using Equation (25).

$$d_j = 1 - e_j \quad (25)$$

Finally, the weights of the attributes can be calculated as follows.

$$w_j = \frac{d_j}{\sum_{j=1}^{n} d_j} \quad (26)$$

3.4.1.2 Weighed Sum Model

Weighed sum model is based on calculating a preference index for each alternative, whereas the best alternative is the one with the highest preference in the maximization case. On the other, the best alternative has the lowest preference in the minimization case. The preference of each alternative can be calculated using Equation (27)

$$P_i = \sum_{j=1}^{n} f_{ij} * w_j \quad (1 \le i \le m, 1 \le j \le n) \tag{27}$$

, where

P_i represents the preference of each alternative. f_{ij} represents the measure of performance in the normalized matrix. w_j represents the weight of each criteria. m and n represent the number of alternatives and the number of criteria, respectively.

3.4.1.3 COPRAS

COPRAS is defined as complex proportional assessment. COPRAS method assumes direct, proportional dependence of significance and priority of investigated alternatives in a system containing attributes. The preference of alternative is calculated taking into account the positive and negative characteristics of alternatives. COPRAS method calculates the utility degree of each alternative as per below procedure (Mulliner et al., 2013). The normalization process can be performed using Equation (28).

$$d_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x_{ij}} q_j \tag{28}$$

, where

x_{ij} is the value that corresponds to the measure of performance of the $i-th$ alternative and $j-th$ attribute and q_j represents the weight of each attribute. d_{ij} represents dimensionless weighted value. The weights of attributes can be calculated using Equation (29).

$$q_j = \sum_{i=1}^{m} d_{ij} \tag{29}$$

The alternatives are distinguished by beneficial (maximizing) attributes and cost (minimizing) attributes. The sum of weighted normalized values for both the beneficial and cost attributes can be obtained using Equations (30) and (31), respectively.

$$s+_i = \sum_{j=1}^{n} d_{ij} \tag{30}$$

$$s-_i = \sum_{j=1}^{n} d_{ij} \tag{31}$$

, where

$s+_i$ refers to sum of elements in the weighted normalized matrix that corresponds to beneficial attributes. On the other hand, $s-_i$ refers to sum of elements in the weighted normalized matrix that corresponds to cost attributes. The relative significance (Q_i) is calculated for each alternative using Equation (32).

$$Q_i = s+_i + \frac{s-_{min} * \sum_{i=1}^{m} s-_i}{s-_i * \sum_{i=1}^{m} (s-_{min}/s-_i)} = s+_i + \frac{\sum_{i=1}^{m} s-_i}{s-_i * \sum_{i=1}^{m} (1/s-_i)} \tag{32}$$

The utility degree of each alternative is calculated and the best alternative is the alternative with the highest utility degree. The utility degree for each alternative is computed using Equation (33).

$$N_j = \frac{Q_i}{Q_{max}} * 100\% \tag{33}$$

, where

N_j indicates the utility degree of each alterative.

3.4.1.4 VIKOR

The Serbian name for VIKOR technique is "Vlse Kriterijumska Optimizacija I Resenje" which means multi criteria optimization and compromise solution. VIKOR is divided into five steps (Cristóbal, 2011):

The first step is to determine the best and worst values for all criteria which depends whether the attributes are cost or beneficial. The best and worst values for the beneficial and cost attributes can be calculated using Equations (34) and (35), respectively.

$$f_j* = \max f_{ij},\ f_{j-} = \min f_{ij} \mid j\ associated\ with\ benefit\ criteria \tag{34}$$

$$f_j* = \min f_{ij},\ f_{j-} = \max f_{ij} \mid j\ associated\ with\ cost\ criteria \tag{35}$$

The second step is to calculate $S_i,\ R_i$. The terms $S_i,\ R_i$ represent utility measure and regret measure, respectively. The utility measure and regret measure are computed using Equations (36) and (37), respectively.

$$S_i = \sum_{j=1}^{n} (w_j * \frac{f_j* - f_{ij}}{f_j* - f_j-}) \tag{36}$$

$$R_i = max_j (w_j * \frac{f_j* - f_{ij}}{f_j* - f_j-}) \tag{37}$$

The third step is to calculate Q_i. Q_i is calculated using Equation (38). Ʊ represents the weight of the decision making strategy or maximum group utility. The term $(1 - Ʊ)$ represents the weight of the individual regret. The value Ʊ is usually taken 0.5 and its value is between 0 and 1. The overall ranking index (Q_i) can be computed using Equation (38).

$$Q_i = Ʊ * \frac{S_i - S*}{S_- - S*} + (1 - Ʊ) * \frac{R_i - R*}{R_- - R*} \tag{38}$$

, where

$S* = min_i S_i,\ S\_ = max_i S_i$

$R* = min_i R_i,\ R\_ = max_i R_i$

The fourth step is to sort alternatives by the value of S, R and Q in decreasing order. The fifth step is to propose as compromise solution Alternative (A_1) which is best ranked by Q and it has the minimum Q if the following two conditions are satisfied:

1. Acceptable advantage where $Q(A_2)$- $Q(A_1)$>= DQ where $DQ = 1/(n-1)$ and A_2 is the second alternative ranked by Q and n represents the number of alternatives.
2. Acceptable stability in decision making where alternative (A_1) must be best ranked by S and/or R. The compromise solution should be stable within decision making process which could be the strategy of maximum of group utility when U > 0.5 or by consensus when U =0.5 or by veto when U <0.5.

If one of the conditions is not satisfied, then a set of compromise solutions are proposed which are:

1. Alternative A_1, A_2 if only the second condition is not satisfied.
2. Alternatives $A_1, A_2, A_3, \ldots\ldots\ldots\ldots A_M,$ if condition 1 is not satisfied. A_M is determined by the relation $Q(A_M)$- $Q(A_1)$ < DQ for maximum M. The positions of these alternatives are "in closeness".

3.4.1.5 Grey Relational Analysis

The process of grey relational analysis (GRA) is divided into four main steps which are (Kuo et al., 2008):

Grey Relational generating is normalization process for performance attributes. Equation (39) is used to normalize beneficial attributes (the higher value the better option). Equation (40) is used to normalize non-beneficial attributes (the lower value the better option). Equation (41) is used to normalize attributes where the closer to the desired value ($x_j *$) the better option.

$$y_{ij} = \frac{x_{ij} - min\{x_{ij}, i = 1,2, \dots \dots \dots m\}}{max\{x_{ij}, i = 1,2, \dots \dots \dots m\} - min\{x_{ij}, i = 1,2, \dots \dots \dots m\}} \quad (39)$$

$$y_{ij} = \frac{max\{x_{ij}, i = 1,2, \dots \dots \dots m\} - x_{ij}}{max\{x_{ij}, i = 1,2, \dots \dots \dots m\} - min\{x_{ij}, i = 1,2, \dots \dots \dots m\}} \quad (40)$$

$$y_{ij} = \frac{|x_{ij} - x_j *|}{max\{x_{ij}, i = 1,2, \dots \dots \dots m\} - min\{x_{ij}, i = 1,2, \dots \dots \dots m\}} \quad (41)$$

Reference sequence generation is the second step where the performance values are defined within the range [0, 1]. For the cost category, it is the lowest value while in the benefit category, it is the highest value. Grey Relational coefficient generation is the third step. The aim of this step is to determine whose compatibility sequence is closest to the reference sequence. Grey relational coefficient is calculated using Equation (42).

$$\gamma(y_{0j}, y_{ij}) = \frac{\Delta_{min} + \xi\Delta_{max}}{\Delta_{ij} + \xi\Delta_{max}} \quad (42)$$

, where

$\Delta_{ij} = |y_{0j} - y_{ij}|$

$\Delta_{min} = min\{\Delta_{ij}, i = 1,2, \dots \dots \dots m; j = 1,2, \dots \dots \dots n\}$

$\Delta_{max} = max\{\Delta_{ij}, i = 1,2, \dots \dots \dots m; j = 1,2, \dots \dots \dots n\}$

ξ is the distinguishing coefficient and it is within the range [0, 1].

The grey relational grade is calculated using Equation (43). The best alternative is the alternative with the highest relational grade.

$$r(y_0, y_i) = \sum_{j=1}^{n} w_j * \gamma(y_{0j}, y_{ij}) \quad (43)$$

3.4.1.6 TOPSIS

TOPSIS stands for Technique for Order Preference by Similarity to Ideal Solution. TOPSIS utilizes the Euclidean distances to compare between the alternatives using the positive and negative ideal solutions as a reference. TOPSIS decision making technique is divided into five main steps (Dragia et al., 2013):

The decision matrix is normalized where the purpose of this step is to convert the performance attributes into non-dimensional ones. The normalized decision matrix is computed using Equation (44).

$$r_{ij} = \frac{x_{ij}}{\sum_{i=1}^{m} x^2{}_{ij}} \quad (44)$$

, where

x_{ij} represents the measure of performance of the $i-th$ alternative with respect to $j-th$ attribute.

The weighted normalized matrix is obtained using Equation (45).

$$v_{ij} = r_{ij} * w_j \quad (45)$$

, where

w_j represents the weight of the $j-th$ attribute.

The ideal and negative ideal solutions are determined. $A*$ indicates the most preferable alternative or ideal solution. On the contrary, $A-$ indicates the least preferable alternative or negative ideal solution. For benefit criteria, decision maker wants to obtain the maximum value among all alternatives. On the other hand, the decision maker wants to obtain minimum value among all alternatives for cost criteria. The ideal solution and negative ideal solution can be computed using Equations (46) and (47), respectively.

$$A*=\left\{\left(max\ v_{ij}\middle|j\in J\right),\left(min\ v_{ij}\middle|j\in J'\right),i=1,2,3,\ldots\ldots\ldots M\right\}=\{v*_1,v*_2\ldots\ldots\ldots v*_N\} \quad (46)$$

$$A-=\left\{\left(min\ v_{ij}\middle|j\in J\right),\left(max\ v_{ij}\middle|j\in J'\right),i=1,2,3,\ldots\ldots\ldots M\right\}=\{v-_1,v-_2\ldots\ldots\ldots v-_N\} \quad (47)$$

Such that;

$$J=\{,j=1,2,3,\ldots\ldots\ldots N|j\ associated\ with\ benefit\ criteria\}$$

$$J'=\{,j=1,2,3,\ldots\ldots\ldots N|j\ associated\ with\ cost\ criteria\}$$

, where

M represents the number of alternatives. N represents the number of attributes.

The fourth step is to calculate the separation distance of each alternative to the ideal and negative ideal solutions. $s*_i$ represents the separation distance of each alternative in the Euclidean way from the ideal solution. On the contrary, $s-_i$ represents the separation distance of each alternative in the Euclidean way from the negative ideal solution. The separation distance to the

ideal solution and the separation distance to the negative ideal solution can be computed using Equations (48) and (49), respectively.

$$s*_i = \left(\sum_{j=1}^{n}(v_{ij} - v*_j)\right)^{\frac{1}{2}} \quad (48)$$

$$s-_i = \left(\sum_{j=1}^{n}(v_{ij} - v-_j)\right)^{\frac{1}{2}} \quad (49)$$

The fifth step is to calculate the relative closeness of an alternative A_i to the ideal solution A $*$. The relative closeness is calculated using Equation (50). when $c*_i$ is close, this means that the solution is closer to the ideal solution. Alternatives are ranked in descending order.

$$c*_i = \frac{s-_i}{s*_i + s-_i} \quad (50)$$

3.4.2 Group Decision-Making

Each one of the multi-criteria decision-making techniques provides a distinct ranking for the candidate solutions. Thus, group decision-making is performed to aggregate the results using Equation (51). The group decision-making helps the decision makers to extract the best compromise solution. Group decision-making is performed based on the genetic algorithm by minimizing the foot rule distances obtained from the five multi-criteria decision-making techniques and the designated ranking for the different candidate solutions. This methodology provides the most feasible ranking that better simulates the ranking of the decision-making techniques. The alternative with the first ranking is the alternative that represents the thresholds of the GPR scale.

$$F = min \sum_{t=1}^{T}\sum_{i=1}^{I}(||R_{it} - R||) \quad (51)$$

, where

R_{it} represents the ranking obtained for the $i-th$ alternative by the $t-th$ multi-criteria decision making technique. R represents the designated ranking for the alternatives. I and T indicate number of alternatives and multi-criteria decision making techniques, respectively.

4 MODEL IMPLEMENTATION

The proposed methodology is implemented for four bridge decks in North America: three of them are in Quebec, Canada, and one of them is in New Jersey, United States. The four bridges are denoted as bridge "A", bridge "B", bridge "C", and bridge "D". The signals were collected using a GSSI 1.5GHZ antenna. All the calculations and optimization algorithms took place on a 2.6 GHZ Intel laptop. GSSI RADAN7 software is used to pick the amplitude values of the top reinforcing rebars as shown in Figure 9.

INSERT FIGURE 9

RapidMiner 7.5 is one of the platforms that are used to perform the clustering algorithms. The clustering model is divided into eight main sub modules where the clustering algorithm is performed using the" clustering" submodule (see Figure 10). The number of optimization steps is assumed 100 for all the clustering algorithms. The clusters obtained from the expectation maximization clustering algorithm are shown in Figure 11. "C" label represents the amplitude values of bridge "A". Cluster 0 represents the "good" category. Cluster 1 represents the "medium" category. Cluster 2 represents the "very severe" category. Cluster 3 represents the "severe" category. Thresholds obtained from the expectation maximization clustering algorithm of bridge "A" are -25.538, -10.964, and -2.767, respectively.

INSERT FIGURE 10

INSERT FIGURE 11

Fuzzy C-means is one of the used clustering algorithms in the proposed methodology. The fuzzifier constant is assumed four, and the number of iterations is assumed 100. The degrees of membership of a sample of the amplitude values are shown in Table 1. Each data point is assigned to the cluster that has the maximum degree of membership. A sample of the thresholds obtained from some of the clustering algorithms is shown in Table 2. As shown in Table 2, the thresholds obtained from the clustering algorithms are different within the same bridge, e.g., threshold 1is -17.158 decibels for the kernel k-means in Bridge "C" while threshold 1 is -14.058 decibels for the expectation maximization in Bridge "C". The thresholds obtained from the same clustering algorithm of two bridges are different, e.g., threshold 1 obtained from the fuzzy c-means in Bridge "C" is -10.214 decibels while threshold 1 of the fuzzy c-means of bridge "A" is -15.706 decibels.

INSERT TABLE 1

INSERT TABLE 2

Due to the difference in the thresholds obtained from the clustering algorithms, the multi-objective optimization module is performed based on the four objective functions defined in the "Model Development" section. In order to provide a fair comparison between the optimization algorithms, 20 independent optimization runs are carried out with different initializations for the multi-objective shuffled frog leaping algorithm, multi-objective particle swarm algorithm and multi-objective genetic algorithm. The number of iterations is assumed 200, and the population size is assumed 250 for all the evolutionary algorithms to provide an equal basis of comparison. For the genetic algorithm, tournament selection is the parent selection strategy. Two-point crossover is utilized, and the crossover rate is assumed 0.8. Mutation rate is assumed 0.1. For the

particle swarm optimization, the cognitive learning and social parameters are assumed two. The inertia weight is assumed 0.5.

For the shuffled frog leaping algorithm, the number of memeplexes is assumed 25, i.e., 10 frogs per each memeplex. The convergence of the SFL is shown in Figure 12. Figure 12 illustrates the behavior of the clustering index objective function through a number of iterations. The objective function begins to stabilize starting from the iteration 180. A sample of the optimum solutions obtained from the SFL is shown in Table 3. SFL generated very promising results when compared to the genetic algorithm, and the particle swarm optimization algorithm.

INSERT FIGURE 12

INSERT TABLE 3

A sample of the Pareto frontier points for one of the runs is shown in Figure 13 and Figure 14. Twenty seven Pareto frontier points are obtained from the three evolutionary algorithms, i.e., 12 points from the shuffled frog leaping algorithm, 10 points from the genetic algorithm, and 5 points from the particle swarm algorithm. The black bubbles, blue bubbles, and red bubbles represent the Pareto frontier points of the shuffled frog leaping algorithm, genetic algorithm, and particle swarm algorithm, respectively. As shown in Figures 13 and 14, SFL generates the most feasible optimal solutions. However, a further detailed analysis is conducted to compare between the optimization algorithms

INSERT FIGURE 13

INSERT FIGURE 14

A comparison between shuffled frog-leaping, particle swarm algorithm and genetic algorithm based on the output of the 20 runs is illustrated in Table 4. The numbers mentioned herein represent the average values. Shuffled frog-leaping algorithm achieved the lowest objective function value regarding objective functions 1, 2 and 3 while the genetic algorithm had the lowest objective function value regarding the objective function 4. The worst objective function value of shuffled frog-leaping algorithm is better than other evolutionary algorithms for the first three objective functions. However, genetic algorithm has a better worst objective function value for the fourth objective function. The mean value obtained employing SFL is better than the other two algorithms regarding the first three objective functions while GA achieved the best mean value for the fourth objective function. SFL has the lowest standard deviation in terms of the four objective functions. A lower standard deviation indicates higher stability of the algorithm while a higher mean value indicates more accuracy of the optimization algorithm.

GA has the lowest coefficient of variation for the first objective function while SFL has the lowest coefficient of variation for the remaining three objective functions. SFL has the largest hypervolume indicator (84.87%) followed by particle swarm optimization algorithm and finally the genetic algorithm. In terms of the inverted generational distance, the SFL has the least inverted generational distance (0.0034) when compared to other algorithms. For the processing time, the average processing times of the shuffled frog leaping algorithm, particle swarm algorithm and genetic algorithm are 131.97 minutes, 97.048 minutes and 88.143 minutes. Thus, SFL has the longest average computational time while genetic algorithm has the shortest average computational time.

Two-tailed Student's t-tests were performed to evaluate the significance level of the optimal solutions, whereas the significance level (α) is set to be 0.05. The paired Two-tailed Student's t-

tests for the three investigated optimization algorithms are depicted in Table 5. The performed student's t-tests examine the null hypothesis (H_0), which is that there is no significant difference between the optimal solutions obtained from each pair of meta-heuristic optimization algorithms. On the other hand, the alternative hypothesis (H_1) assumes that there is a significant difference between the optimal solutions obtained from each pair of the of meta-heuristic optimization algorithms. If the $P-value$ is less than the significance level, then the null hypothesis is rejected in favor of the alternative hypothesis. Nevertheless, if the $P-value$ is more than the significance level, thus the null hypothesis is accepted. As shown in Table 5, the $P-values$ of the pairs (shuffled frog leaping algorithm, particle swarm algorithm), (shuffled frog leaping algorithm, genetic algorithm) and (particle swarm algorithm, genetic algorithm) are less than 0.05, which means that the null hypothesis (H_0) is false. Thus, there is a statistical significant difference between the optimal solutions obtained from the optimization algorithms. The results also illustrates that the performance of the SFL is statistically significantly better than the PSO and the performance of the PSO is statistically significantly better than the GA, and the optimal solutions of the SFL are very statistically significantly better than the GA. Based on the previous statistics, SFL significantly outperformed PSO and GA.

INSERT TABLE 4

INSERT TABLE 5

The decision-making module is implemented to select the best solution among the Pareto frontier points obtained from the multi-objective optimization module. There are four attributes which are: $RMSE_1$, $RMSE_2$, $RMSE_3$, and CLU whereas the weights of the attributes are calculated based on the Shannon entropy method. The weights of the four attributes ($RMSE_1$, $RMSE_2$,

$RMSE_3$, and CLU) are 22.11%, 27.23%, 34.83%, and 15.82%, respectively. The calculations of the weights of the attributes are illustrated in Table 6. A sample of the solution ranking obtained from the TOPSIS is depicted in Table 7. Each one of the decision-making provided a distinct ranking for the solutions. For instance, TOPSIS selected the solution [-16.7619, -8.8161, -2.9744] as the best solution. On the other hand, COPRAS selected the solution [-16.729, -8.8339, -2.9727] as the best solution. Thus, group decision-making is essential to aggregate the rankings obtained from the several multi-criteria decision-making techniques based on a single objective optimization problem (Equation (51)). Genetic algorithm is implemented where the population size and the number of generations are assumed 100. The crossover rate, and the mutation rate are assumed 0.8, and 0.1, respectively. Based on the decision-making module, the standardized thresholds are: -16.7619, -8.8161, and -2.9744 decibels.

INSERT TABLE 6

INSERT TABLE 7

A corrosion map is developed for a bridge that is located on the Chemin Saint-Grégoire in municipality Les Cèdres that overpasses Autoroute 20, Quebec, Canada. The bridge was constructed in 1965 with a total length of 212 feet, and the width of the bridge decks is 42 feet. As shown in Figure 15, the area percentages of the "good", "medium", "severe", and "very severe" categories are: 45.78%, 34.26%, 12.98%, and 6.98%, respectively. The corrosion index (CI) is 77.6% based on Equation (1) which indicates that the bridge deck is in the "medium" category.

INSERT FIGURE 15

5 CONCLUSION

Bridge Management Systems play a very important role in managing large transportation networks where condition assessment is considered as one of the pillars of BMSs. Ground penetrating radar is one of the non-destructive techniques that are utilized to evaluate the corrosion of steel reinforcement in bridge decks. However, the absence of standardized thresholds for the amplitude values is one of the drawbacks of utilizing the ground penetrating radar. In this paper, a hybrid model is implemented in order to calculate the standardized thresholds for the amplitude values for any number of bridge condition categories. The hybrid model is divided into three main modules which are: 1) clustering module. 2) optimization module, and 3) decision-making module. The clustering module incorporates a mixture of soft and hard clustering algorithms such as K-means, fast K-means, kernel K-means, K-medoids, expectation maximization, fuzzy C-means, X-means, and agglomerative clustering. The optimization module calculates the standardized thresholds based on four objective functions that combine both the local search and the global search. The optimization module utilizes three evolutionary algorithms which are: genetic algorithm, particle swarm algorithm, and shuffled frog-algorithm.

A detailed comparison among the three evolutionary algorithms is presented using some performance metrics such as hypervolume indicator, inverted generational distance, coefficient of variation, student's t-test, etc. Decision-making module is used to calculate the most feasible solution among the optimum solutions obtained from the optimization module. Five multi-criteria decision-making techniques are investigated which are: WSM, COPRAS, VIKOR, GRA, and TOPSIS. Each one of the multi-criteria decision-making techniques provides a distinct ranking for the alternative. Consequently, group decision-making is implemented to provide a

final ranking for the alternatives based on a single-objective optimization function. The standardized thresholds obtained from the proposed methodology are: -16.7619, -8.8161, and -2.9744 decibels.

Acknowledgment

This project was funded by the Academy of Scientific Research and Technology (ASRT), Egypt, JESOR-Development Program - Project ID: 40.

Compliance with ethical standards

Conflict of interest: The authors declare that they have no conflict of interest.

Ethical approval: The article does not contain any studies with human participants or animals performed by any of the authors.

6 REFERENCES

1. AASHTO Highways subcommittee on Bridges and Structures (2011) The Manual for Bridge Evaluation: Second Edition. American Association of State Highway and Transportation Officials, Washington, D.C, United States
2. Akyene T (2012) Cell Phone Evaluation Base on Entropy and TOPSIS. Interdisciplinary Journal of Research In Business 1(12): 9–15.
3. Baltar A.M, Fontane D.G (2008) Use of Multiobjective Particle Swarm Optimization in Water Resources Management. Journal of Water Resources Planning and Management 134 (3): 257–265.
4. Barnes C.L, Trottier J.-F, Forgeron D (2008) Improved Concrete Bridge Deck Evaluation Using GPR by Accounting For Signal Depth– amplitude Effects. NDT&E International

41(6): 427–433.

5. Caldas L.G, Norford L.K (2002) A Design Optimization Tool Based on A Genetic Algorithm. Automation in Construction 11(2): 173–184.
6. Cristóbal S J.R (2011) Multi-criteria Decision-making In The Selection Of A Renewable Energy Project in Spain: The Vikor Method. Renewable Energy 36(2): 498-502. doi:10.1016/j.renene.2010.07.031.
7. Czepiel E. (1995) Bridge Management Systems Literature Review and Search. Northwestern University, United States of America.
8. Davies D.L, Bouldin D.W (1979) A Cluster Separation Measure. IEEE Transactions On Pattern Analysis And Machine Intelligence 1(2): 224–227.
9. Dragisa S., Bojan D, Mira D (2013) Comparative Analysis of Some Prominent MCDM Methods: A Case of Ranking Serbian Banks. Serbian Journal of Management 8(2): 213–241.
10. Dunn J.C (1974) Well-Separated Clusters and Optimal Fuzzy Partitions. Journal of Cybernetics 4(1): 95–104.
11. Dinh K, Zayed T (2016) GPR-Based Fuzzy Model for Bridge Deck Corrosiveness Index. Journal of Performance of Constructed Facilities 30(4): 1–14.
12. Elbeltagi E, Hegazy T, Grierson D (2005) Comparison Among Five Evolutionary-based Optimization Algorithms. Advanced Engineering Informatics 19(1): 43–53.
13. Felio G (2016) Canadian Infrastructure Report Card. Canadian Construction Association, Canadian Public Works Association, Canadian Society for Civil Engineering, and Federation of Canadian Municipalities, Canada.
14. Fornell C (1983) Issues in the Application of Covariance Structure Analysis: A Comment. Journal of Consumer Research 9(4): 443–448.

15. Fu G, Kapelan Z, Reed P (2012) Reducing the Complexity of Multi-Objective Water Distribution System Optimization Through Global Sensitivity Analysis. Journal of Water Resources Planning and Management 138(3).
16. Golden Software LLC (2014) Surfer12 software. http://www.goldensoftware.com/products/surfer. Accessed 05 June 2016.
17. Grant T, Urszula C, Reershemius G, Pollard D, Hayes S, Plappert G (2017) Quantitative Research Methods for Linguists: A Questions and Answers Approach for Students, Routledge, United Kingdom.
18. Heidari E, Movaghar A (2011) An Efficient Method Based On Genetic Algorithm To Solve Sensor Network Optimization Problem. International Journal On Applications Of Graph Theory In Wireless Ad hoc Networks And Sensor Networks 3(1): 18–33.
19. Ishibuchi H, Hiroyuki M, Yuki T, Yusuke N (2014) Difficulties in Specifying Reference Points to Calculate the Inverted Generational Distance for Many-Objective Optimization Problems. In: 2014 IEEE Symposium on Computational Intelligence in Multi-Criteria Decision-Making (MCDM), Florida, United States of America, pp 1-14.
20. Keskin G.A (2015) Using Integrated Fuzzy DEMATEL and Fuzzy C : Means Algorithm For Supplier Evaluation and Selection. International Journal of Production Research 53(12): 3586-3602.
21. Mashwani K.N, Salhi A (2016) Multiobjective Evolutionary Algorithm Based on Multimethod with Dynamic Resources Allocation. Applied Soft Computing 39: 292–309. doi:10.1016/j.asoc.2015.08.059.
22. Kumar D.N, Reddy M.J (2007) Multipurpose Reservoir Operation Using Particle Swarm Optimization. Journal Of Water Resources Planning And Management 133(3): 192–201.

23. Kuo Y, Yang T, Huang G.W (2008) The Use of Grey Relational analysis In solving multiple attribute decision-making problems. Computers and Industrial Engineering 55(1): 80–93. doi:10.1016/j.cie.2007.12.002.
24. KNIME Company (2016) KNIME 3.3.1 software. https://www.knime.org/. Accessed 06 August 2016.
25. Mackenzie H (2013) Canada's Infrastructure Gap: Where It Came from and Why It Will Cost So Much to Close. Canadian Centre for Policy Alternatives, Canada.
26. MathWorks (2013) Matlab R2013a software. https://www.mathworks.com/. Accessed 06 Septemebr 2014.
27. Martino N, Maser K, Birken R, Wang M (2016) Quantifying Bridge Deck Corrosion Using Ground Penetrating Radar. Research in Nondestructive Evaluation 27(2): 112-124.
28. Mirza S (2007) Danger Ahead: The Coming Collapse of Canada's Municipal Infrastructure. Federation of Canadian Municipalities, Canada.
29. Mulliner E, Smallbone K, Maliene V (2013) An Assessment Of Sustainable Housing Affordability Using A Multiple Criteria Decision Making Method. Omega 41(2): 270–279.
30. Nebro A.J, Durillo J.J, Coello C.A.C (2013) Analysis of Leader Selection Strategies In A Multi-Objective Particle Swarm Optimizer. In: 2013 IEEE Congress on Evolutionary Computation, Cancún, México, pp 3153-3160.
31. Orouji H, Mahmoudi N, Pazoki M, Biswas A (2016) Shuffled Frog-leaping Algorithm For Optimal Design Of Open Channels. Journal of Irrigation and Drainage Engineering 142(10).
32. Pourbahman Z, Hamzeh A (2015) A Fuzzy Based Approach for Fitness Approximation in Multi-Objective Evolutionary Algorithms. Journal of Intelligent and Fuzzy Systems 29: 2111-2131.

33. RapidMiner Inc (2016) RapidMiner 7.5 software. https://rapidminer.com/products/studio/. Accessed 06 August 2016.
34. Riquelme N, Lucken C, Baran B (2015) Performance Metrics in Multi-Objective Optimization. In: 2015 XLI Latin American Computing Conference (CLEI), Peru, pp 1-11.
35. Sawant K.B (2015) Efficient Determination of Clusters in K-Mean Algorithm Using Neighborhood Distance. International Journal of Emerging Engineering Research and Technology 3(1): 22–27.
36. Sahani R, Bhuyan P.K (2014) Pedestrian Level Of Service Criteria For Urban Off-street Facilities In Mid-Sized Cities. Transport 32(2): 221-232.
37. Shami A (2015) Ground Penetrating Radar-based Deterioration Assessment of Bridge Decks. M.Sc thesis, Concordia University, Canada.
38. Statistics Canada (2014) From Roads to Rinks: Government Spending on Infrastructure in Canada, 1961 to 2005. http://www.statcan.gc.ca/pub/11-010-x/00907/10332-eng.htm. Accessed 20 Decemeber 2016.
39. Statistics Canada (2009a) Age of Public Infrastructure: A Provincial Perspective. http://www.statcan.gc.ca/pub/11-621-m/11-621-m2008067-eng.htm. Accessed 20 Decemeber 2016.
40. Statistics Canada (2009b) Average age of public infrastructure by province and type of infrastructure, 2007. http://www.statcan.gc.ca/pub/11-621-m/2008067/tables/5002061-eng.htm#archived. Accessed 20 Decemeber 2016.
41. Venkatesan T, Sanavullah M.Y (2013) SFLA Approach To Solve PBUC Problem With Emission Limitation. Electrical Power and Energy System 46: 1–9.
42. Wang L, Fang C (2011) An Effective Shuffled Frog-leaping Algorithm For Multi-mode

Resource-constrained Project Scheduling Problem. Information Sciences 181(20): 4804–4822.

43. Yang I (2007) Using Elitist Particle Swarm Optimization to Facilitate Bicriterion Time-cost Trade-off Analysis. Journal of Construction Engineering and Management 133(7): 498–505.
44. Zhang H, Li H (2010) Multi-objective Particle Swarm Optimization For Construction Time-cost Tradeoff Problems. Construction Management and Economics 28(1): 75–88.

List of Figures

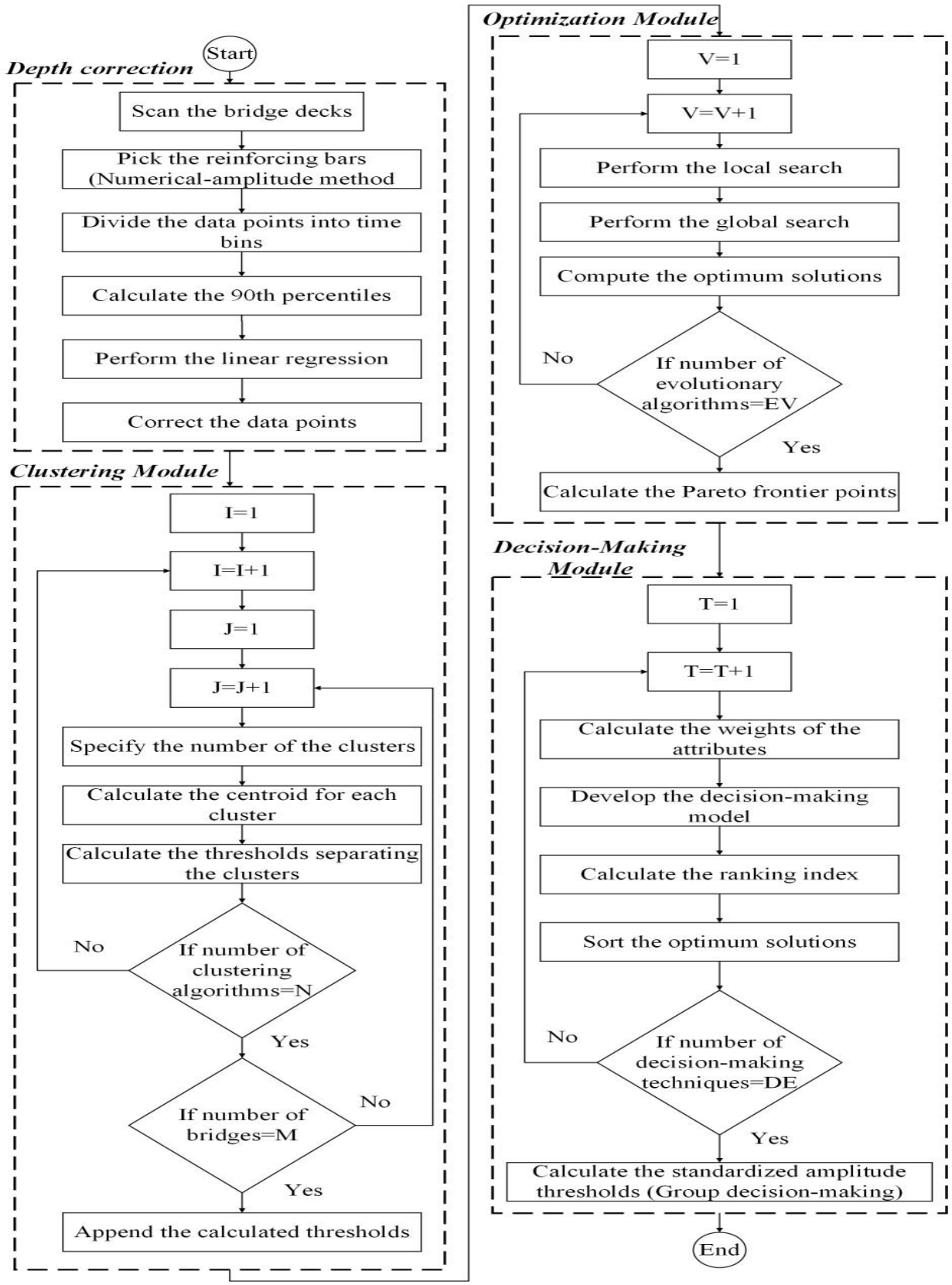


Figure 1: Flowchart of the proposed methodology

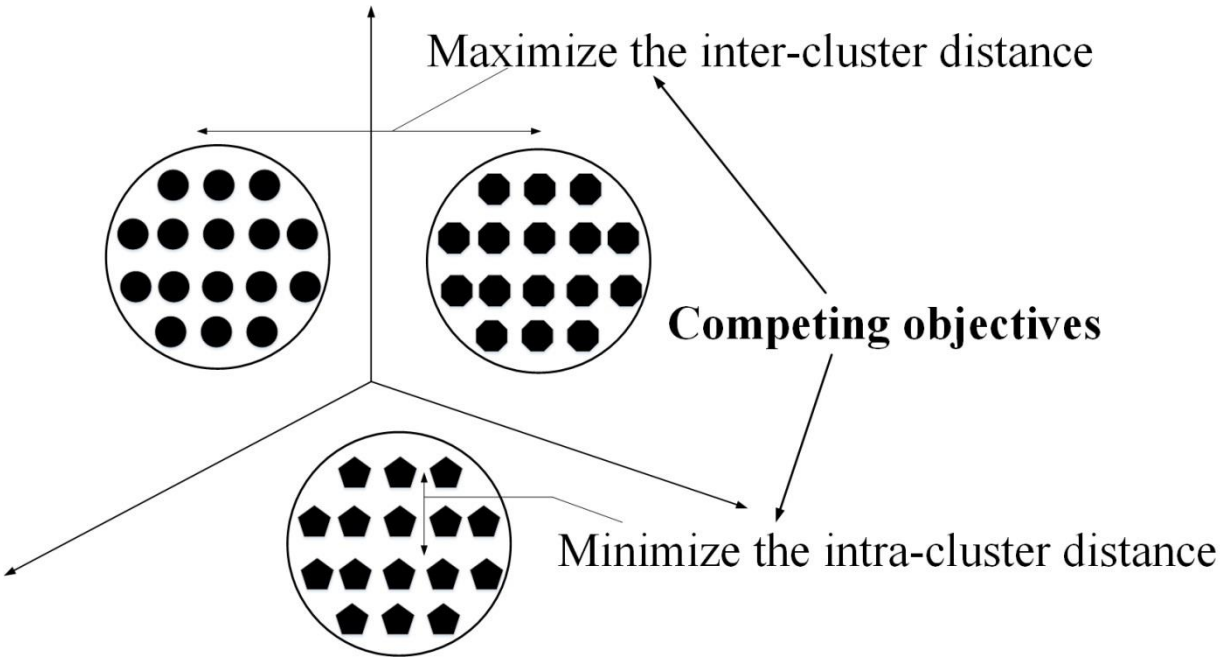


Figure 2: Representation of inter-cluster and intra-cluster distances

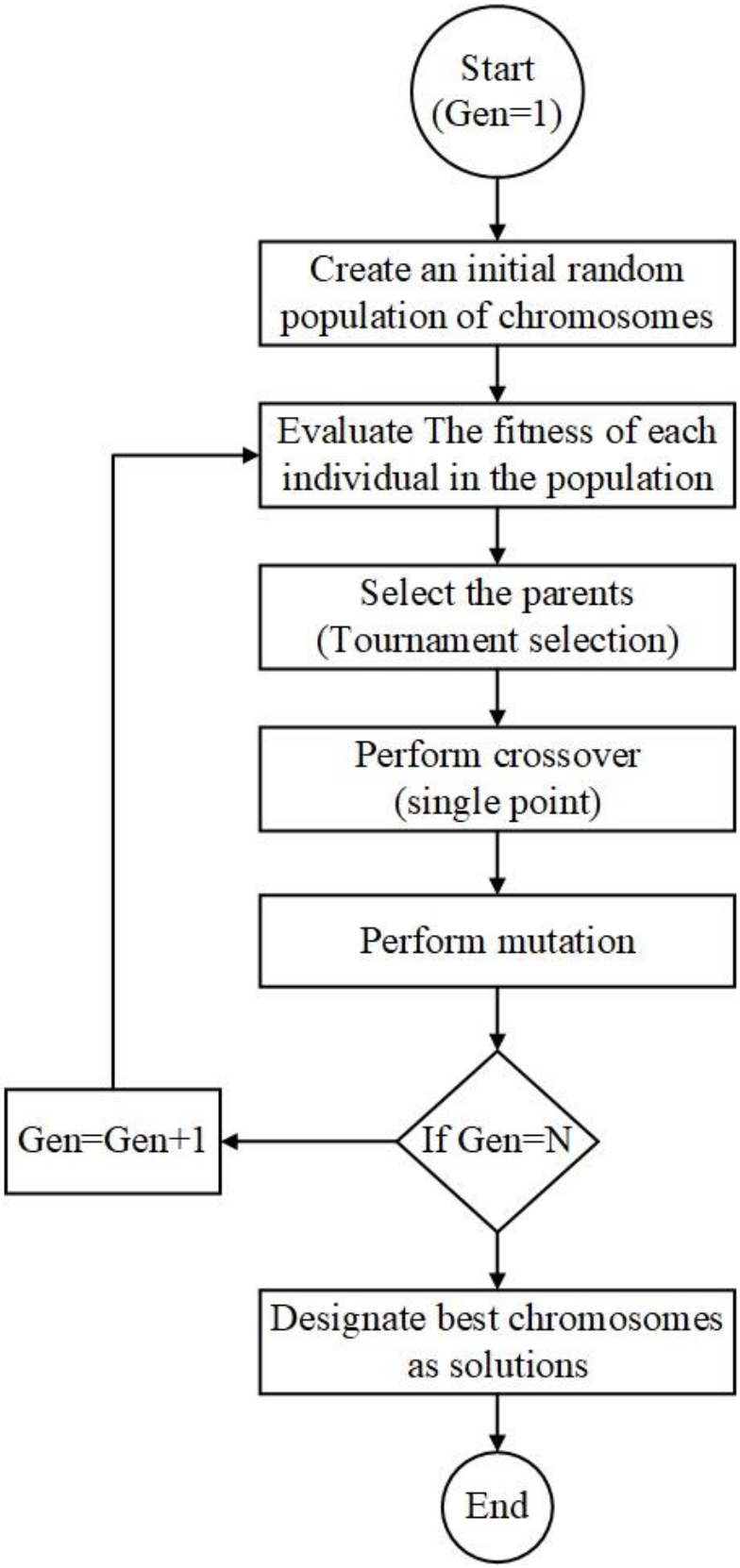


Figure 3: Flowchart of the genetic algorithm

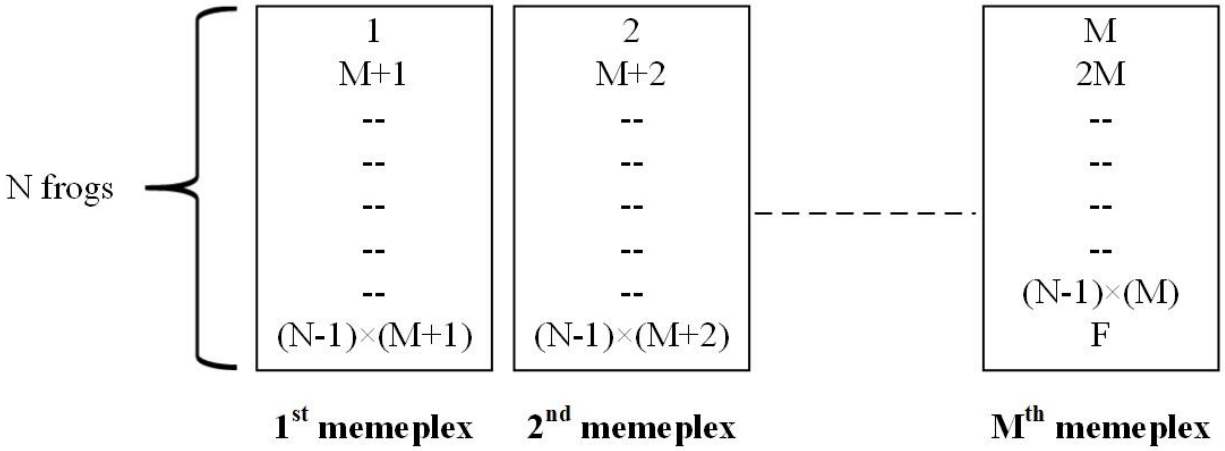


Figure 4: Basic concept of the shuffled frog leaping algorithm

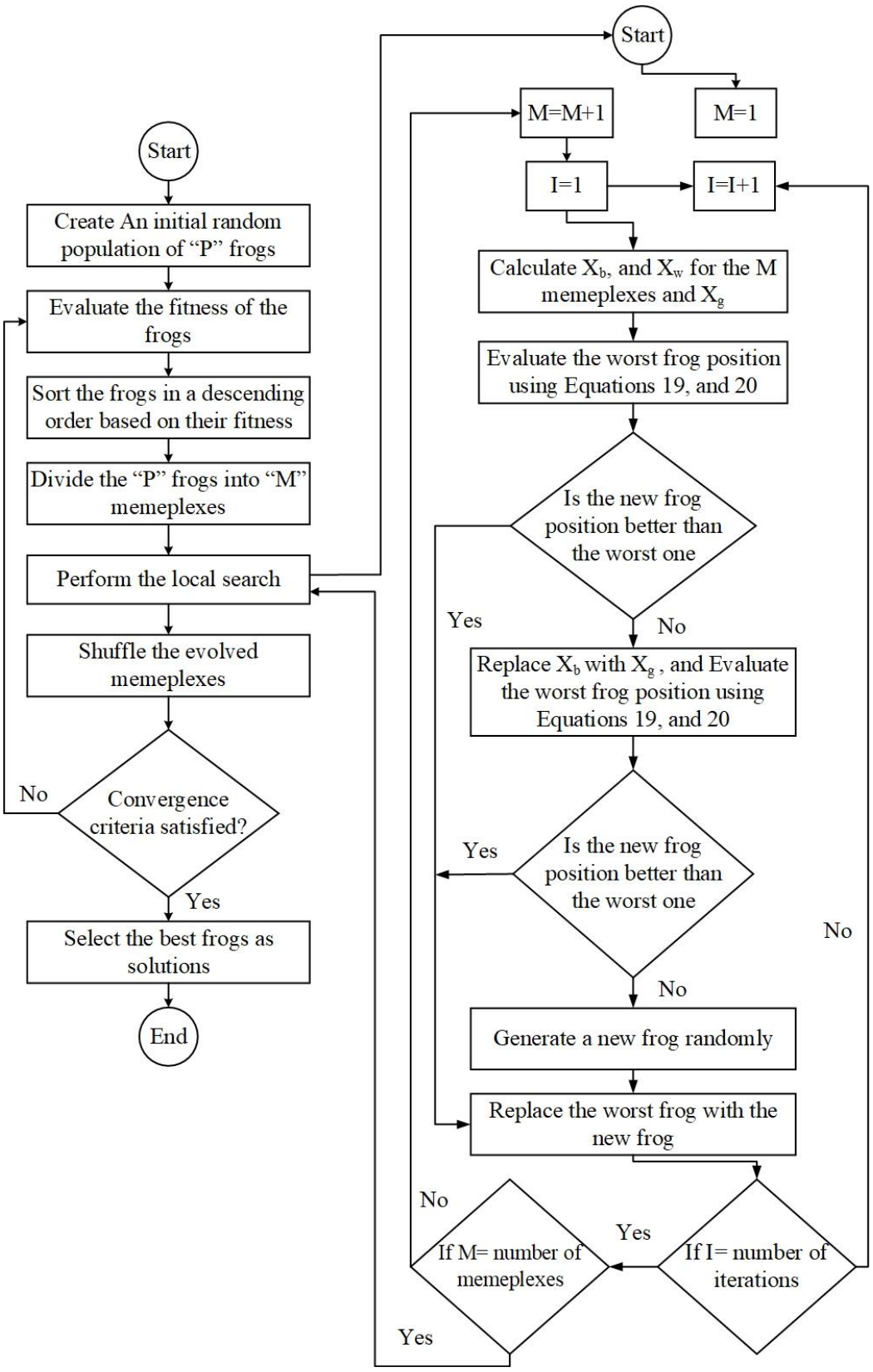


Figure 5: Flowchart of shuffled frog leaping algorithm

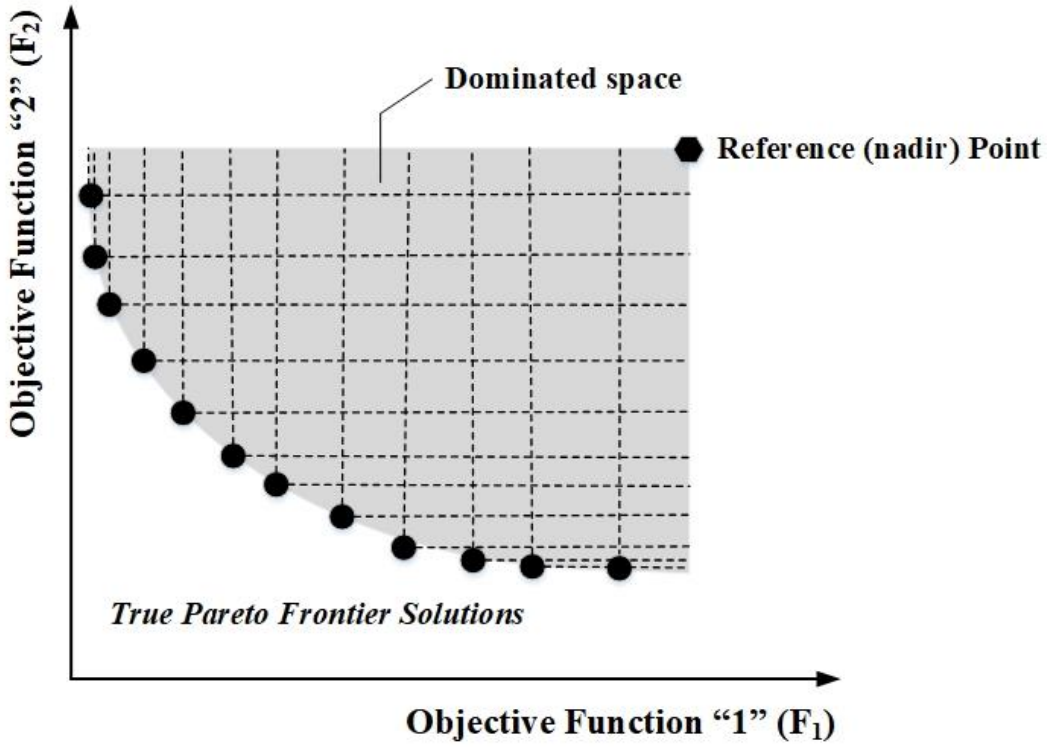


Figure 6: Visualization of the hypervolume indicator

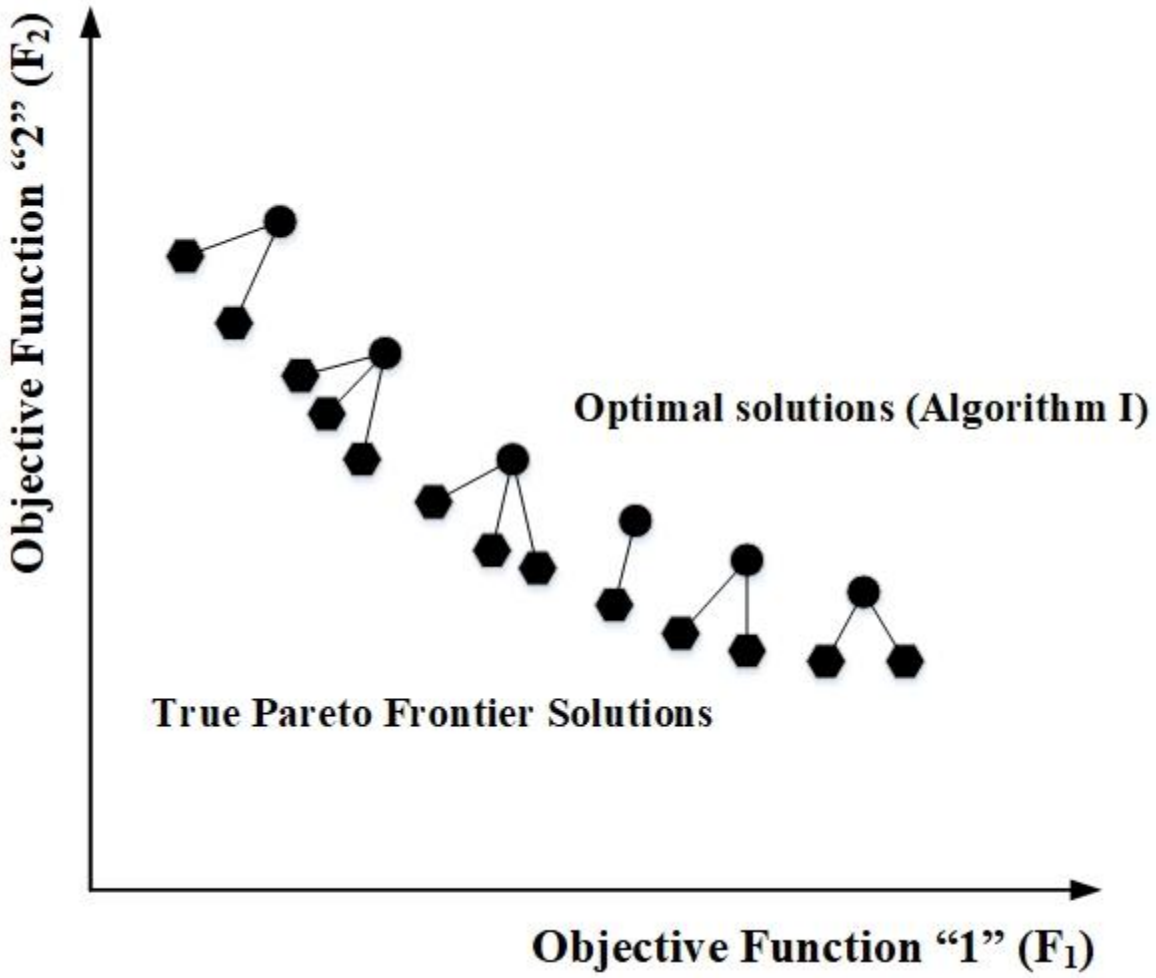


Figure 7: Visualization of inverted general distance metric

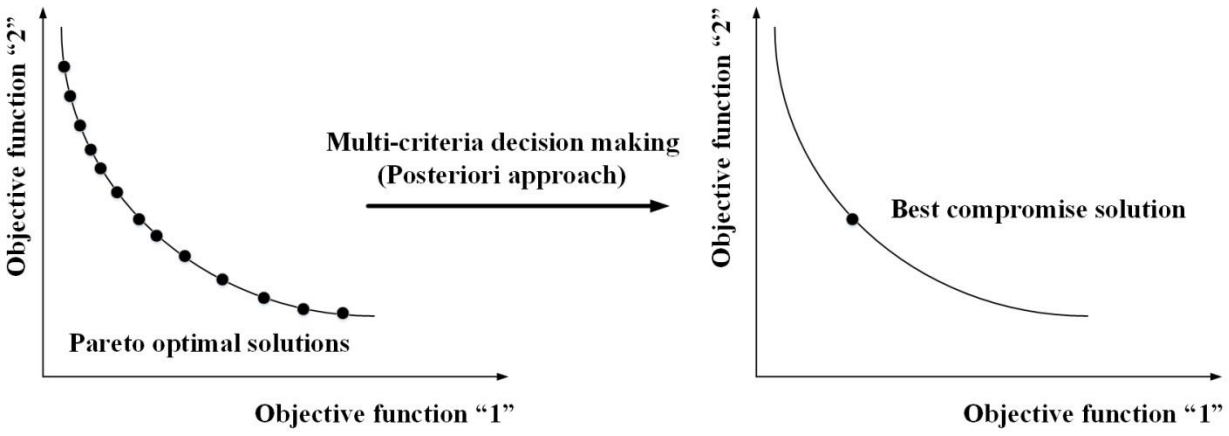


Figure 8: Hybrid optimization-decision making model

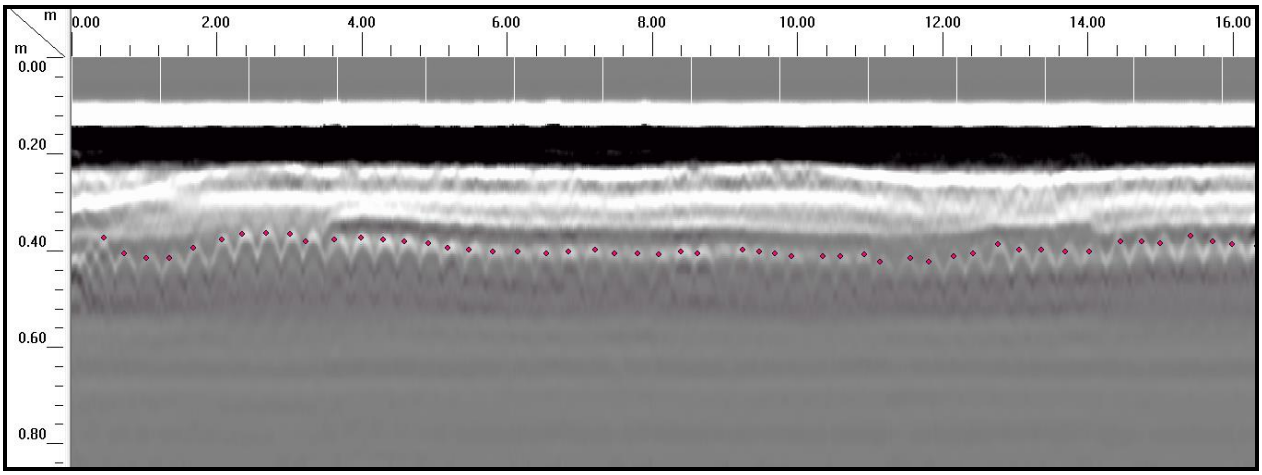


Figure 9: Picking the amplitude values of the top reinforcing rebars

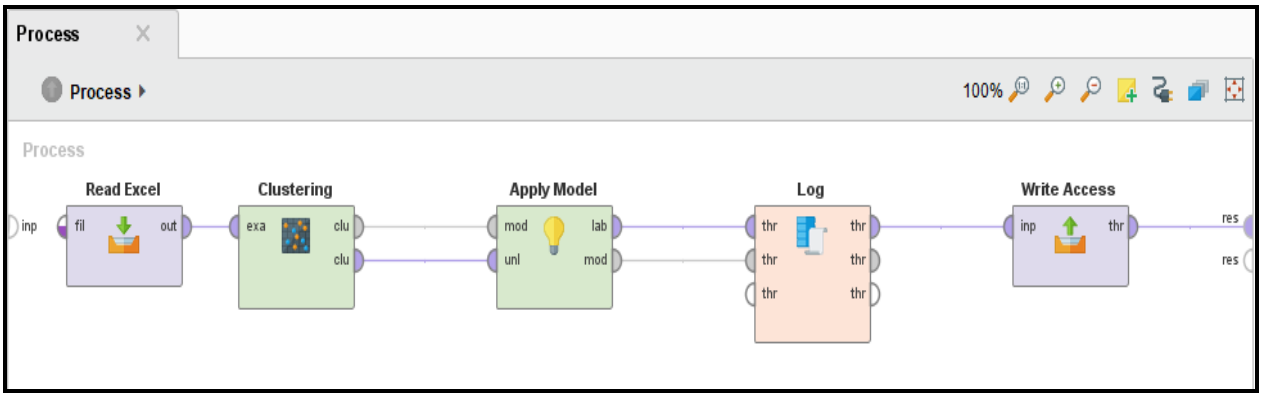


Figure 10: Interface of the RapidMiner platform

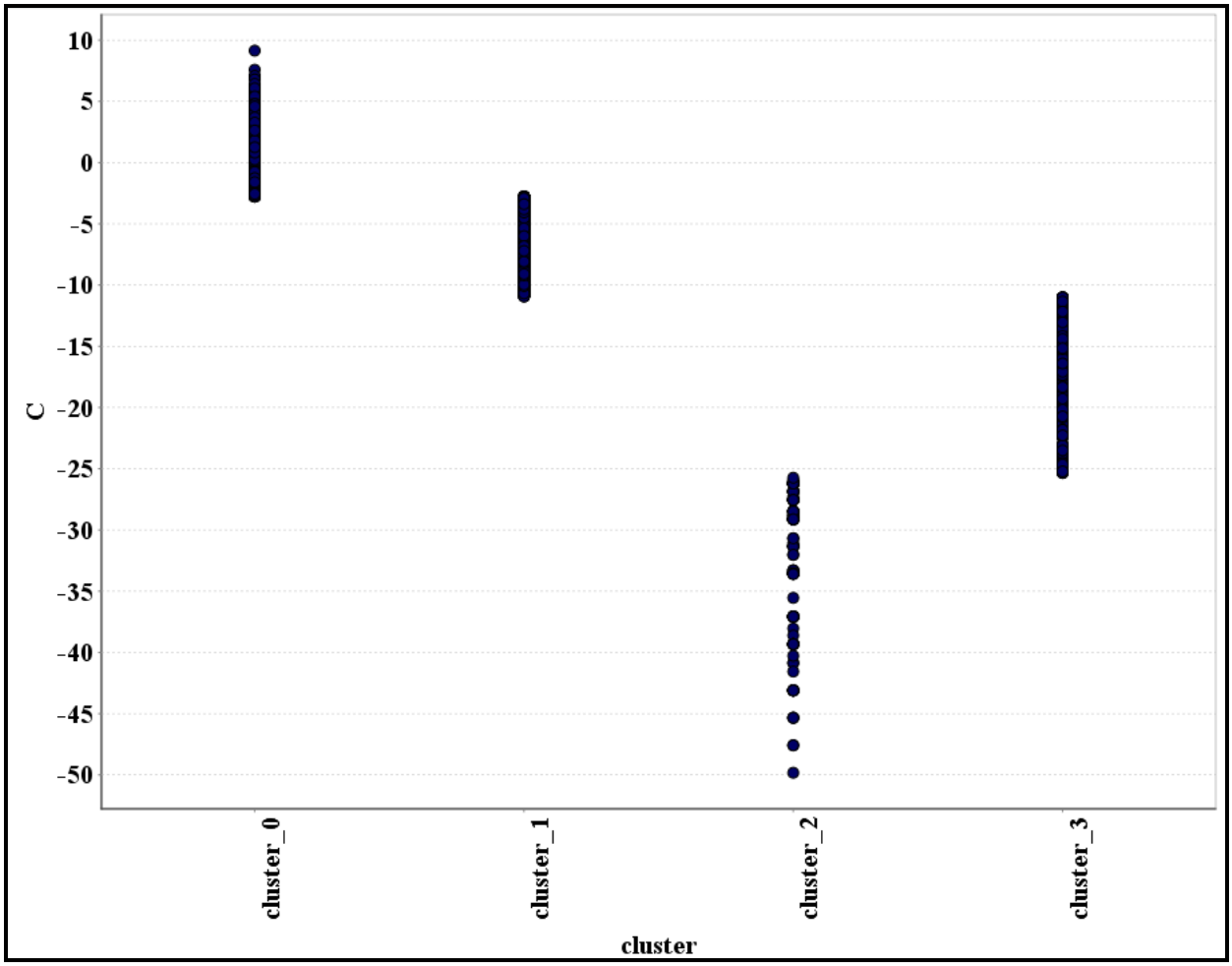


Figure 11: Clusters obtained from expectation maximization clustering algorithm

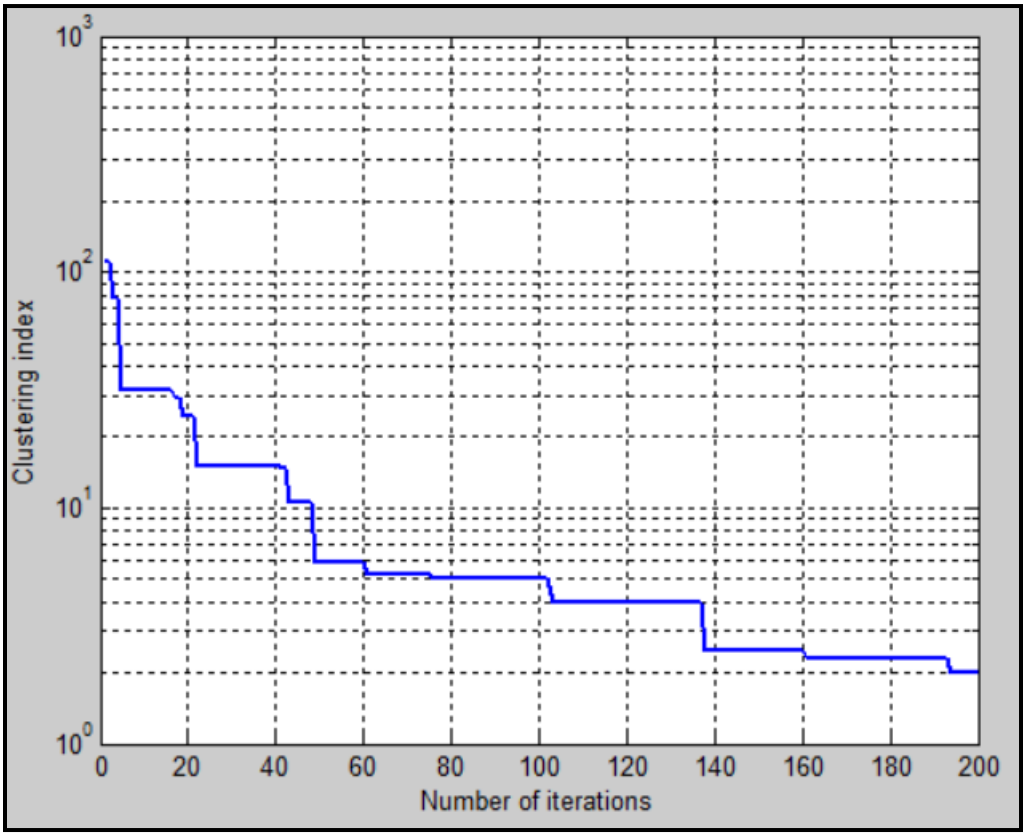


Figure 12: Convergence of the shuffled frog leaping algorithm

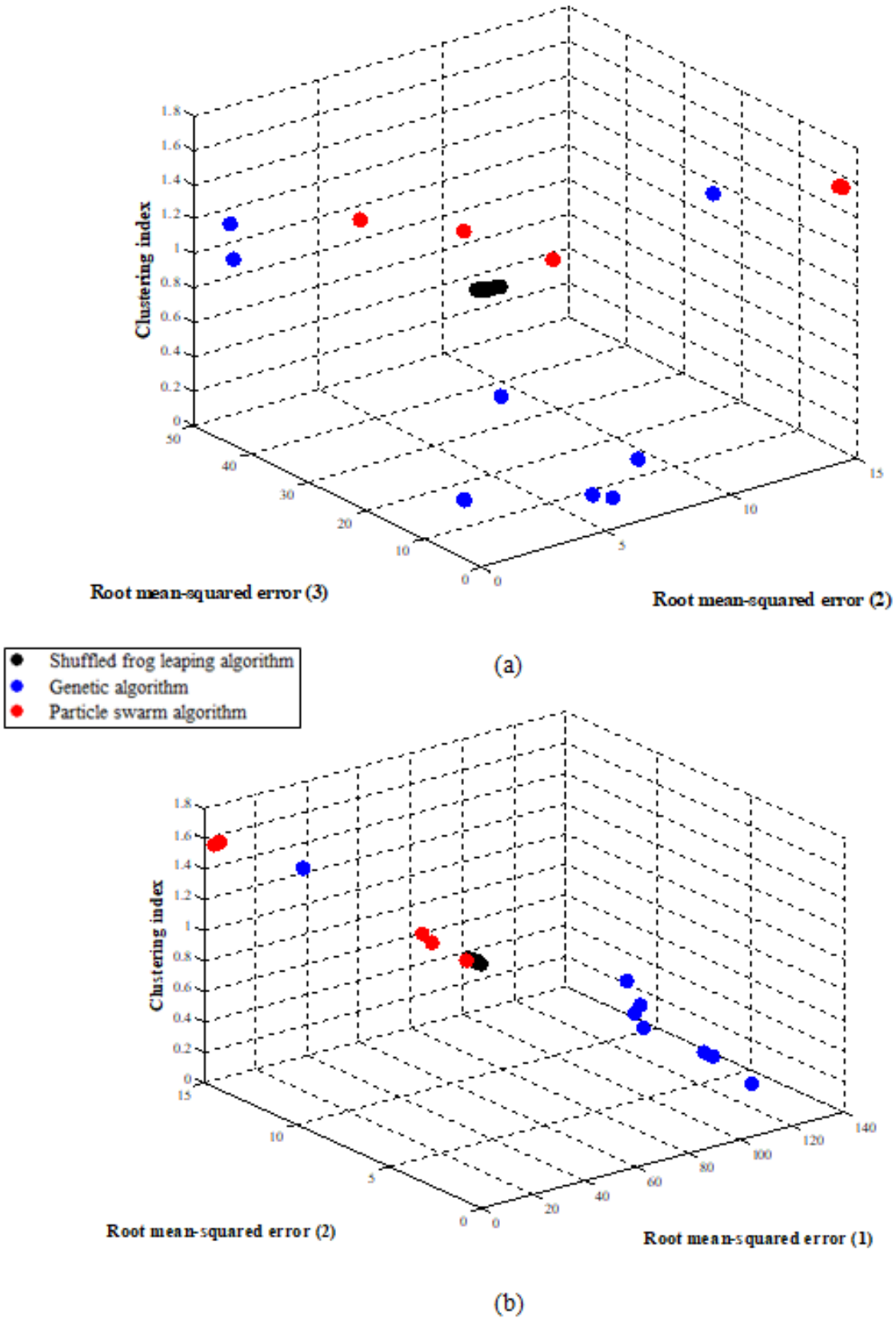


Figure 13: Pareto frontier points of the three evolutionary algorithms- A

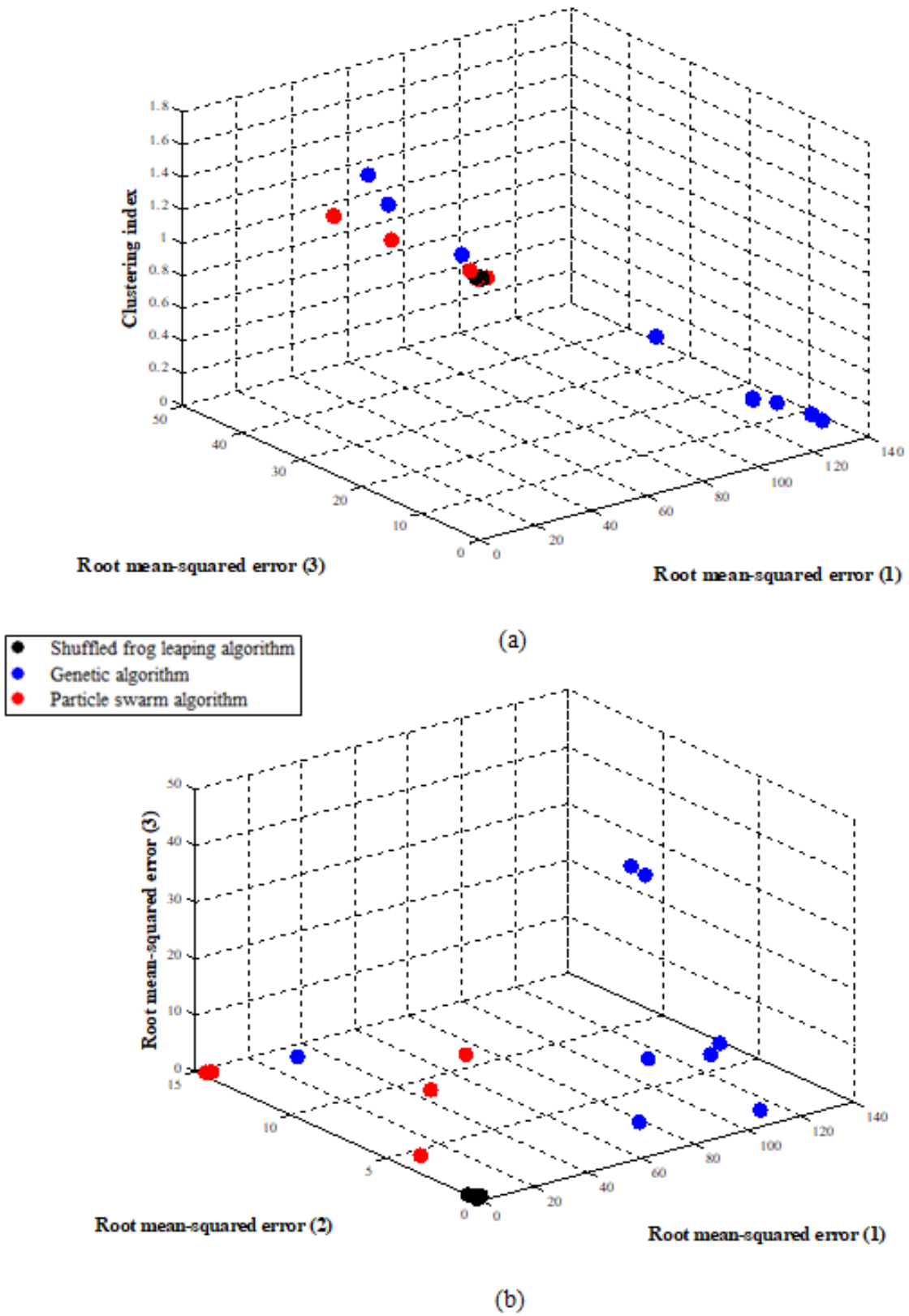
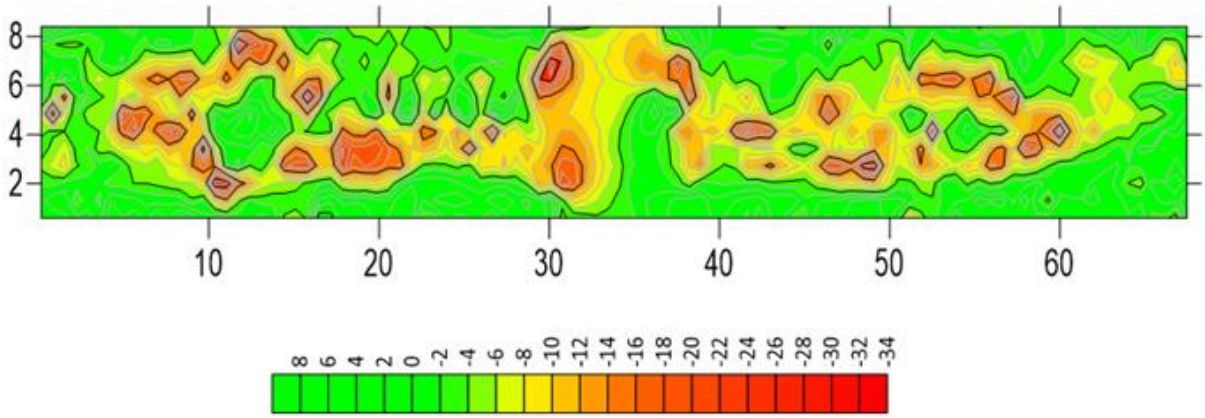


Figure 14: Pareto frontier points of the three evolutionary algorithms- B



Good	Medium	Severe	Very severe
45.78%	34.26%	12.98%	6.98%

Figure 15: Developed corrosion map for Bridge "A"

List of Tables

Table 1: Sample of the cluster memberships obtained from the fuzzy C-means algorithm

Data point	Degree of membership				Assigned cluster
	Cluster 0	Cluster 1	Cluster 2	Cluster 3	
-1.422	0.349	0.16	0.403	0.086	Cluster 2
-2.09	0.42	0.166	0.326	0.086	Cluster 0
-1.689	0.376	0.164	0.371	0.087	Cluster 0
-0.839	0.29	0.146	0.481	0.081	Cluster 2
-1.13	0.32	0.154	0.44	0.084	Cluster 2
0.318	0.128	0.073	0.754	0.042	Cluster 2

Table 2: A sample of the thresholds obtained from some clustering algorithms

Clustering algorithm	Bridge number	Threshold 1	Threshold 2	Threshold 3
Expectation maximization	Bridge" D"	-8.101	-3.24	-0.206
Kernel K-means	Bridge "C"	-17.158	-12.318	-8.173
Expectation maximization	Bridge "C"	-14.058	-6.693	-0.752
Fuzzy C-means	Bridge "C"	-10.214	-6.003	-1.699
K-means	Bridge "A"	-22.736	-10.505	-2.83
Expectation maximization	Bridge "A"	-25.538	-10.946	-2.767
Fuzzy C-means	Bridge "A"	-15.706	-6.752	-1.66
Fuzzy C-means	Bridge "B"	-13.963	-6.296	-2.701

Table 3: Sample of the optimal solutions of the shuffled frog leaping algorithm

Optimal Solutions (decibels)	Objective function "1" ($RMSE_1$)	Objective function "2" ($RMSE_2$)	Objective function "3" ($RMSE_3$)	Objective function "4" (CLU)
[-16.6637, -8.7673, -2.9887]	0.5281	0.5306	0.0493	1.6003
[-16.729, -8.8339, -2.9277]	0.1644	0.1597	0.3885	1.6008
[-16.7505, -8.882, -2.8774]	0.0446	0.1082	0.6887	1.6001
[-16.8005, -8.8227, -3.0706]	0.2334	0.2223	0.4068	1.6018
[-16.7911, -8.8046, -2.9266]	0.1814	0.323	0.3947	1.6001
[-16.8363, -8.8094, -2.9632]	0.433	0.2962	0.1913	1.6018
[-16.7611, -8.8676, -2.844]	0.0423	0.0282	0.8545	1.6
[-16.6637, -8.7673, -2.9887]	0.5281	0.5306	0.0493	1.6003

Table 4: Comparison between shuffled frog-leaping, particle swarm algorithm and genetic algorithm for the twenty runs

Index	Objective function	Shuffled frog-leaping	Particle swarm optimization	Genetic algorithm
Minimum	Objective function "1"	**0.0187**	0.0423	65.4994
	Objective function "2"	**0.0282**	0.8257	0.0044
	Objective function "3"	**0.0019**	0.3135	1.3756
	Objective function "4"	1.5998	1.5804	**0.1321**
Maximum	Objective function "1"	**0.6957**	3.0568	127.4886
	Objective function "2"	**0.7561**	14.5401	10.9924
	Objective function "3"	**0.8545**	24.6486	49.2263
	Objective function "4"	1.6018	1.6036	**1.5843**
Mean	Objective function "1"	**0.2598**	1.2496	89.1582
	Objective function "2"	**0.3177**	7.2220	3.2343
	Objective function "3"	**0.3259**	8.7193	12.3022
	Objective function "4"	1.6006	1.5907	**0.6197**
Standard deviation	Objective function "1"	**0.196**	1.1312	34.197
	Objective function "2"	**0.2312**	6.0043	3.5267
	Objective function "3"	**0.2755**	9.6845	18.1101
	Objective function "4"	**0.0008**	0.0081	0.4647
Coefficient of variation	Objective function "1"	0.7546	0.9052	**0.3836**
	Objective function "2"	**0.7280**	0.8314	1.0904
	Objective function "3"	**0.8453**	1.1107	1.4721
	Objective function "4"	**0.0005**	0.0051	0.7498
Hypervolume indicator (HV)	…….	**84.87%**	70.65%	50.58%
Inverted generational distance (IGD)	…….	**0.0034**	0.011	0.0037
Computational time (minutes)		131.97	97.0482	**88.143**

Table 5: Statistical comparison between optimization algorithms based on two-tailed Student's t-test

Pair of optimization algorithm	Shuffled frog leaping algorithm	Particle swarm algorithm	Genetic algorithm
Shuffled frog leaping algorithm	H_0 ($P-value$=1)	H_1 ($P-value$ =4.235×10^{-5})	H_1 ($P-value$ =2.649×10^{-6})
Particle swarm algorithm	H_1 ($P-value$ =4.235×10^{-5})	H_0 ($P-value$ =1)	H_1 ($P-value$ =3.994×10^{-5})
Genetic algorithm	H_1 ($P-value$ =2.649×10^{-6})	H_1 ($P-value$ =3.994×10^{-5})	H_0 ($P-value$ =1)

Table 6: Entropy values, variation coefficients, and the weights of the attributes

Index	$RMSE_1$	$RMSE_2$	$RMSE_3$	CLU
Entropy value (e_j)	0.476	0.355	0.175	0.625
variation coefficient (d_j)	0.523	0.6447	0.824	0.374
weights of the attribute (w_j)	22.11%	27.23%	34.83%	15.82%

Table 7: Sample of the solutions' ranking obtained from TOPSIS

Solution (decibels)	Evolutionary algorithm	s_i^*	s_i^-	c_i^*	Solution ranking
[-16.7619, - 8.8161, -2.9744]	SFL	0.03318	0.28816	0.89673	1
[-16.729, -8.8339, -2.9727]	SFL	0.03315	0.28773	0.89669	2
[-16.705, -8.882, -2.8774]	SFL	0.03321	0.287303	0.89636	4
[-29.2791, -8.5982, -3.5222]	GA	0.03397	0.28581	0.89377	17
[-35.5542, -8.8618, -3.5443]	GA	0.0332	0.28798	0.89661	15
[-29.1454, -8.6766, -5.5362]	GA	0.227	0.1491	0.3964	26
[-17.3076, -6.2511, -2.9412]	PSO	0.15622	0.2444	0.61006	24
[-16.7866, -9.0109, 1.4294]	PSO	0.11914	0.206	0.63356	21
[-16.751, -6.2662, -3.1229]	PSO	0.15537	0.2435	0.6105	23