

Volume 44 Number 1 March 2020

ISSN 0350-5596

# *Informatica*

**An International Journal of Computing  
and Informatics**



1977

## Editorial Boards

Informatika is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor from the Editorial Board can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the list of referees. Each paper bears the name of the editor who appointed the referees. Each editor can propose new members for the Editorial Board or referees. Editors and referees inactive for a longer period can be automatically replaced. Changes in the Editorial Board are confirmed by the Executive Editors.

The coordination necessary is made through the Executive Editors who examine the reviews, sort the accepted articles and maintain appropriate international distribution. The Executive Board is appointed by the Society Informatika. Informatika is partially supported by the Slovenian Ministry of Higher Education, Science and Technology.

Each author is guaranteed to receive the reviews of his article. When accepted, publication in Informatika is guaranteed in less than one year after the Executive Editors receive the corrected version of the article.

### Executive Editor – Editor in Chief

Matjaž Gams  
Jamova 39, 1000 Ljubljana, Slovenia  
Phone: +386 1 4773 900, Fax: +386 1 251 93 85  
matjaz.gams@ijs.si  
<http://dis.ijs.si/mezi/matjaz.html>

### Editor Emeritus

Anton P. Železnikar  
Volaričeva 8, Ljubljana, Slovenia  
s51em@lea.hamradio.si  
<http://lea.hamradio.si/~s51em/>

### Executive Associate Editor - Deputy Managing Editor

Mitja Luštrek, Jožef Stefan Institute  
mitja.lustrek@ijs.si

### Executive Associate Editor - Technical Editor

Drago Torkar, Jožef Stefan Institute  
Jamova 39, 1000 Ljubljana, Slovenia  
Phone: +386 1 4773 900, Fax: +386 1 251 93 85  
drago.torkar@ijs.si

### Executive Associate Editor - Deputy Technical Editor

Tine Kolenik, Jožef Stefan Institute  
tine.kolenik@ijs.si

### Editorial Board

Juan Carlos Augusto (Argentina)  
Vladimir Batagelj (Slovenia)  
Francesco Bergadano (Italy)  
Marco Botta (Italy)  
Pavel Brazdil (Portugal)  
Andrej Brodnik (Slovenia)  
Ivan Bruha (Canada)  
Wray Buntine (Finland)  
Zhihua Cui (China)  
Aleksander Denisiuk (Poland)  
Hubert L. Dreyfus (USA)  
Jozo Dujmović (USA)  
Johann Eder (Austria)  
George Eleftherakis (Greece)  
Ling Feng (China)  
Vladimir A. Fomichov (Russia)  
Maria Ganzha (Poland)  
Sumit Goyal (India)  
Marjan Gušev (Macedonia)  
N. Jaisankar (India)  
Dariusz Jacek Jakóbczak (Poland)  
Dimitris Kanellopoulos (Greece)  
Samee Ullah Khan (USA)  
Hiroaki Kitano (Japan)  
Igor Kononenko (Slovenia)  
Miroslav Kubat (USA)  
Ante Lauc (Croatia)  
Jadran Lenarčič (Slovenia)  
Shiguo Lian (China)  
Suzana Loskovska (Macedonia)  
Ramon L. de Mantaras (Spain)  
Natividad Martínez Madrid (Germany)  
Sando Martinčić-Ipišić (Croatia)  
Angelo Montanari (Italy)  
Pavol Návrat (Slovakia)  
Jerzy R. Nawrocki (Poland)  
Nadia Nedjah (Brasil)  
Franc Novak (Slovenia)  
Marcin Paprzycki (USA/Poland)  
Wiesław Pawłowski (Poland)  
Ivana Podnar Žarko (Croatia)  
Karl H. Pribram (USA)  
Luc De Raedt (Belgium)  
Shahram Rahimi (USA)  
Dejan Raković (Serbia)  
Jean Ramaekers (Belgium)  
Wilhelm Rossak (Germany)  
Ivan Rozman (Slovenia)  
Sugata Sanyal (India)  
Walter Schempp (Germany)  
Johannes Schwinn (Germany)  
Zhongzhi Shi (China)  
Oliviero Stock (Italy)  
Robert Trappl (Austria)  
Terry Winograd (USA)  
Stefan Wrobel (Germany)  
Konrad Wrona (France)  
Xindong Wu (USA)  
Yudong Zhang (China)  
Rushan Ziatdinov (Russia & Turkey)

# Improvement of the Deep Forest Classifier by a Set of Neural Networks

Lev V. Utkin and Kirill D. Zhuk

Peter the Great, St.Petersburg Polytechnic University (SPbPU), Russia

E-mail: lev.utkin@gmail.com

**Keywords:** classification, random forest, decision tree, deep learning, neural network, class probability distribution

**Received:** April 1, 2019

*A Neural Random Forest (NeuRF) and a Neural Deep Forest (NeuDF) as classification algorithms, which combine an ensemble of decision trees and neural networks, are proposed in the paper. The main idea underlying NeuRF is to combine the class probability distributions produced by decision trees by means of a set of neural networks with shared parameters. The networks are trained in accordance with a loss function which measures the classification error. Every neural network can be viewed as a non-linear function of probabilities of a class. NeuDF is a modification of the Deep Forest or gcForest proposed by Zhou and Feng, using NeuRFs. The numerical experiments illustrate the outperformance of NeuDF and show that the NeuRF is comparable with the random forest.*

*Povzetek: V povzetku sta predstavljena dva izvorna algoritma: nevronske naključni gozdovi in nevronske globoki gozdovi.*

## 1 Introduction

In spite of the intensive development of a huge number of various modern classification models, including the deep learning models, the ensemble methodology remains one of the most efficient approaches for solving machine learning problems. The ensemble learning models are based on constructing multiple classifiers for training data and on aggregating their corresponding predictions in accordance with a certain rule. The final ensemble classifier is represented as a weighted average of outputs of the base or weak classifiers. The weight of each classifier can be viewed as its contribution to the final decision. Several approaches use some functions that combine the outputs from all base classifiers instead of weighted averages. From a statistical point of view, one of the ideas underlying the improvement of the classifier performance by means of the ensemble combinations is based on reduction of variance of the classification error [11]. This occurs because the usual effect of ensemble averaging is the reduction of the variance of a set of classifiers.

Three main techniques of combining the classifiers can be pointed out [44]: bagging, stacking and boosting. Bagging [4] aims to improve accuracy by combining multiple classifiers. One of the most powerful bagging methods is the random forest (RF) method [5], which uses a large number of individual decision trees in order to combine their predictions. Another technique for achieving the highest generalization accuracy in the framework of ensemble-based methods is stacking [41]. This technique is used to combine various classifiers by means of a meta-learner that takes into account which classifiers are reliable and which are not. The best known ensemble-based technique is boosting which improves the performance of weak

classifiers by means of their combining into a single strong classifier. Both boosting and bagging techniques use voting for combining the classifiers. However, the voting mechanism is differently implemented. In particular, examples in bagging are chosen with equal probabilities. Boosting supposes to choose the examples with probabilities that are proportional to their weights [32].

There are several review papers devoted to various approaches based on the combination of classifiers. A detailed analysis of many ensemble-based methods can be found in a review proposed by Ferreira and Figueiredo [14]. The review compares a huge number of modifications of boosting algorithms. One of the first books thoroughly studying combination rules for improving classification performance was written by Kuncheva [23]. An interesting review of ensemble-based methods is proposed by Polikar [30]. A nice review is presented by Wozniak et al. [42]. A comprehensive analysis of combination algorithms and their application to machine learning approaches such as classification, regression, clusterization can be also found in a review paper written by Rokach [32]. We have to point out also other recent reviews [13, 19, 31, 43]. A detailed description and an exhaustive analysis of most ensemble-based models are given in Zhou's book [44].

One of the widely used and exhibiting extremely high performance ensemble-based methods is a RF [4]. It is a classifier consisting of a collection of randomized decision trees. According to main algorithms for constructing the RF, a certain numbers of training elements and features are drawn at random with replacement from a training set in order to build every decision tree in the forest. The RF models have been successfully used in various practical problems. The detailed descriptions of many RF applications and properties of RFs have been reviewed by many authors

[2, 9, 15, 27, 33].

An interesting new ensemble-based method which can be viewed as a combination of several ensemble-based methods, including the RF and the stacking, is proposed by Zhou and Feng [45] and called the Deep Forest (DF) or gcForest. Its structure consists of layers similar to a multi-layer neural network structure, but each layer in gcForest contains many RFs instead of neurons. gsForest can be regarded as an multi-layer ensemble of decision tree ensembles. As pointed out by Zhou and Feng [45], gcForest is much easier to train and can perfectly work when there are only small-scale training data in contrast to deep neural networks which require great effort in hyperparameter tuning and large-scale training data. A lot of numerical experiments provided by Zhou and Feng [45] illustrated that gcForest outperforms many well-known methods or is at least comparable with them.

Advantages of gcForest motivate us to modify it in order to improve its classification capability. Some improvements have been proposed by Utkin and Ryabinin [37, 38, 39]. In particular, modifications of the DF for solving the weakly supervised and fully supervised metric learning problems were proposed in [39] and [37], respectively. A transfer learning model using the DF was presented in [38]. The main idea underlying the proposed modifications is to assign weights to decision trees in every RF in order to minimize the corresponding loss functions which depend on the problem solved. The weights are used to replace the standard averaging of the class probabilities for every instance and every decision tree with the weighted average. The weights are regarded as training parameters which can be computed by solving the constrained quadratic optimization problems.

By introducing the tree weights, we simultaneously try to overcome another shortcoming of gcForest. It cannot be fully considered as an alternative to deep neural networks due to its uncontrollability in the sense of defining a goal in tasks different from the standard classification. One of the advantages of neural networks is the flexibility of specifying the error or loss function depending on the data processing task or a specific application. The loss function in the standard classification problem is determined by the difference between a true class label of a training set element and a label computed by means of the forward propagation. The Euclidean distance between the input and output of the network is used in autoencoders. Various types of distances between the probability distributions of the source and target data are used in transfer learning problems. The variety of error functions allows solving a lot of machine learning problems by specifying the required loss function. Therefore, another aim of the modifications is to modify gcForest in order to use different loss functions. We have to point out that the idea of weighting in RFs is also not new. Most weighting RF methods use weights of classes to deal with imbalanced datasets, for example, [10]. At the same time, there are a lot of publications devoted to more complex weight assignments to every tree. In partic-

ular, Li et al. [25] propose to assign weights to decision trees according to their classification ability. A similar approach for weighting decision trees is presented by Kim et al. [20]. An interesting study of weighted voting methods in RFs is also given in [34]. The main difference of these methods from the proposed approach is that all the methods use some measures of the classification quality in order to assign the weights. Moreover, these measures are obtained on the basis of testing data. To the best of our knowledge, there are no methods which consider the weights as training parameters. The proposed approach allows us to select a weighting assignment scheme in a flexible way by using different loss functions for optimization.

The approach using weights of decision trees for computing a target probability class vector for every RF have illustrated the outperformance in comparison with gcForest. However, it has some shortcomings. First, the number of weights is strongly depends on the number of decision trees in every RF. On the one hand, we increase the number of trees in order to increase the classification accuracy, but the large number of decision trees leads to the same large number of weights. As a result, the number of training parameters is increased and the model may lead to overfitting. On the other hand, a reduction of decision trees may lead to a reduction of the classification accuracy. Second, the weighted average used for computing the RF probability class vector is a linear function of the weights. This fact significantly restricts a set of possible solutions and may make worse the classifier.

In order to overcome the above difficulties, we propose to use a neural network of a special form for computing the probability class vectors. The neural network plays a role of a non-linear analog of the linear function of weights. Of course, we do not have the weights of decision trees in the explicit form now. But we get a function which combines the probabilities of every class at the leaf nodes in order to obtain the RF probability class vector. In other words, the neural network plays a role of a non-linear function of weights. It should be noted that the proposed neural network is not standard because we have to identically process probabilities of every class. This implies that if the number of classes is  $C$ , then we construct  $C$  identical neural networks with shared parameters. In particular, if a training data have two classes, then the obtained neural network is very similar to the Siamese neural network [6] which has been widely used in many applications (see, for example, [1, 8, 17]). Outputs of all identical networks for every training instance form the corresponding probability class vector. In fact, the neural networks can be viewed as a non-linear alternative to the weighted sum of probabilities. In particular, this approach coincides with the approach using the weighted averages when activation functions of all units in the neural networks are linear. The proposed combinations of the neural network with the RF and the DF are called NeuRF and NeuDF, respectively.

It should be noted that the idea to jointly use RFs and neural networks is not new. An interesting approach for

constructing a denoising RF was proposed by Hibino et al. [16]. Another combination of the RF with the neural network was presented by Kotschieder et al. [21] where an ensemble of random trees is restructured as a collection of random neural networks, which exhibits better generalization performance. The authors of [21] introduced a soft differentiable decision function at the split nodes and a global loss function defined on a tree. Following this approach, several similar models were proposed in [3, 18, 35, 36, 40, 46]. Maji et al. [28] used a deep neural network for unsupervised learning followed by supervised learning of the deep neural network response using a RF.

In contrast to the above combinations of neural networks and RFs, in the presented paper, we incorporate the neural networks into the DF in order to correct and to control the class vectors at outputs of RFs. Our experiments demonstrate that NeuRF and NeuDF are competitive on many publicly available datasets.

## 2 A short introduction to deep forests

One of the important peculiarities of gcForest is its cascade structure proposed by Zhou and Feng [45]. Every cascade is represented as an ensemble of decision tree forests. The cascade structure is a part of a total gcForest structure. It implements the idea of representation learning by means of the layer-by-layer processing of raw features. Each level of cascade structure receives feature information processed by its preceding level, and outputs its processing result to the next level. The architecture of the cascade proposed by Zhou and Feng [45] is shown in Fig. 1. It can be seen from the figure that each level of the cascade consists of several RFs which generate 3-dimensional class vectors concatenated each other and with the original input. It should be noted that this structure of forests can be modified in order to improve the gcForest for a certain application. After the last level, we have the feature representation of the input feature vector, which can be classified in order to get the final prediction. The gcForest representational learning ability is enhanced by applying the second part of gcForest called as the so-called multi-grained scanning. The multi-grained scanning structure uses sliding windows to scan the raw features. Its output is a set of feature vectors produced by sliding windows of multiple sizes. We mainly pay attention to the first part of gcForest because our modification relates to the RFs.

Given an instance, each forest produces an estimate of a class distribution by counting the percentage of different classes of examples at the leaf node where the concerned instance falls into, and then averaging across all trees in the same forest as it is schematically shown in Fig. 2. The class distribution forms a class vector, which is then concatenated with the original vector to be input to the next level of cascade. The usage of the class vector as a result of the RF classification is very similar to the idea un-

derlying the stacking algorithm [41] which trains the first-level learners using the original training dataset. Then the stacking algorithm generates a new dataset for training the second-level learner (meta-learner) such that the outputs of the first-level learners are regarded as input features for the second-level learner while the original labels are still regarded as labels of the new training data. In contrast to the standard stacking algorithm, gcForest simultaneously uses the original vector and the class vectors (meta-learners) at the next level of cascade by means of their concatenation. This implies that the feature vector is enlarged after every cascade level. After the last level, we have the feature representation of the input feature vector, which can be classified in order to get the final prediction. Zhou and Feng [45] propose to use different forests at every level in order to provide the diversity which is an important requirement for the RF construction.

It is interesting to note that the same architecture of the cascade forest was proposed by Miller et al. [29]. This architecture differs from gcForest in using only class vectors at the next cascade levels without concatenation with the original vector. Miller et al. [29] illustrated by numerical experiments that their approach is comparable to the approach [45]. We have to point out that the cascade structure with neural networks without backpropagation instead of forests was proposed by Hettinger et al. [7].

## 3 Weighted averages in forests

One of the ways to improve gcForest is to assign weights to decision trees in every RF. The weights aim to correct the original averaging of class probability distributions over all decision trees in accordance with a predefined objective function. In the standard classification problem, the objective function is the error function or the difference between class labels of training instances and values of the forest class probability distributions. In the metric learning problem, the objective function is the distance between similar and dissimilar instances. Different machine learning problems define the corresponding objective function and the corresponding weights of decision trees.

Our aim is to briefly consider the idea of the weighted average in order to propose the neural networks for processing the class probability distributions. Therefore, we will consider the standard classification problem for simplicity. The classification problem can be formally written as follows. Given  $n$  training data (examples, instances, patterns)  $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\}$ , in which  $\mathbf{x}_i \in \mathbb{R}^m$  represents a feature vector involving  $m$  features and  $y_i \in \{1, \dots, C\}$  represents the class of the associated instances, the task of classification is to construct an accurate classifier  $c : \mathbb{R}^m \rightarrow \{1, \dots, C\}$  that maximizes the probability that  $c(\mathbf{x}_i) = y_i$  for  $i = 1, \dots, n$ .

A decision tree in every forest produces an estimate of the class probability distribution  $p = (p_1, \dots, p_C)$  by counting the percentage of different classes of training examples

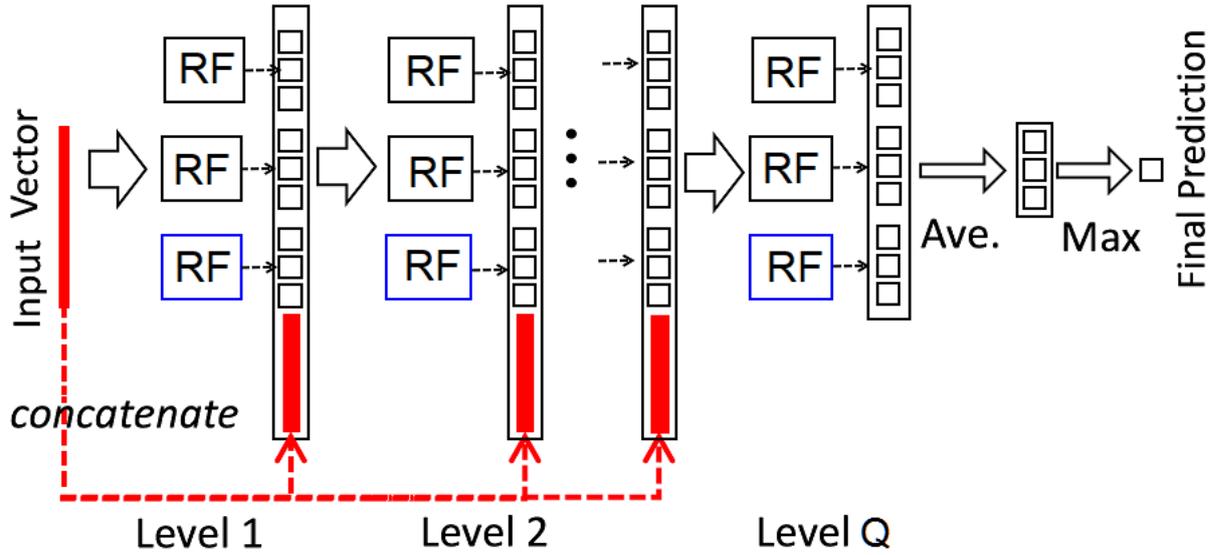


Figure 1: The architecture of the cascade forest [45].

at the leaf node where the concerned instance falls into. Then the class probabilities for every forest are computed by averaging all class probability distributions  $p$  across all trees by taking into account the weights of the trees.

Suppose that all RF have the same number  $T$  of decision trees, every cascade level contains  $M$  RFs, and the number of cascade levels is  $Q$ .

The objective function for computing optimal weights is defined as the Euclidean distance between the class vector and a vector such that its element with index  $y_i$  is 1 and other elements are 0. According to [45], the class distribution forms a class vector which is then concatenated with the original vector to be input to the next level of the cascade. Suppose an origin vector is  $\mathbf{x}_i$ , and the  $p_{i,c}^{(t,k,q)}$  is the probability of class  $c$  for an instance  $\mathbf{x}_i$  produced by the  $t$ -th tree from the  $k$ -th forest at the cascade level  $q$ . Since we consider a single RF at some cascade level, then we omit indices  $k$  and  $q$  corresponding to the forest and the level, respectively. Let us also introduce the notation

$$\begin{aligned} \mathbf{p}_{i,c} &= (p_{i,c}^{(t)}, t = 1, \dots, T), \\ \mathbf{w} &= (w_t, t = 1, \dots, T), \\ \mathbf{v}_i &= (v_{i,c}, c = 1, \dots, C). \end{aligned}$$

Here  $w_t$  is the weight of the  $t$ -th tree in the considered forest. Suppose that  $\mathbf{1}$  is a vector having  $T$  unit elements. Then the  $c$ -th element  $v_{i,c}$  of the class vector produced by the considered forest for the instance  $\mathbf{x}_i$  is determined in gcForest as

$$v_{i,c} = T^{-1} \cdot \mathbf{p}_{i,c} \cdot \mathbf{1}^T \quad (1)$$

The weighted average of the class probability distributions leads to the following class vectors

$$v_{i,c} = \mathbf{p}_{i,c} \cdot \mathbf{w}.$$

It follows from the above that gcForest is a special case of the weighting scheme when all weights are  $1/T$ .

An illustration of the weighted averaging is shown in Fig. 3, where we partly modify a picture from [45] in order to show how elements of the class vector are derived as a simple weighted sum. One can see from Fig. 3 that the augmented features  $v_{i,c}$ ,  $c = 1, \dots, C$ , corresponding to the  $q$ -th forest are obtained as weighted sums, i.e., there hold

$$\begin{aligned} v_{i,1} &= 0.4w_1 + 0.2w_2 + 1.0w_3 + 0.0w_4, \\ v_{i,2} &= 0.4w_1 + 0.5w_2 + 0.0w_3 + 0.0w_4, \\ v_{i,3} &= 0.2w_1 + 0.3w_2 + 0.0w_3 + 1.0w_4. \end{aligned}$$

The weights are restricted by the following obvious conditions:

$$\mathbf{w} \cdot \mathbf{1}^T = 1, w_t \geq 0, t = 1, \dots, T. \quad (2)$$

Now we can write the objective function for computing optimal weights:

$$J(\mathbf{w}) = \min_{\mathbf{w}} \sum_{i=1}^n \|\mathbf{v}_i - \mathbf{o}_i\|_2^2 + \lambda R(\mathbf{w}).$$

Here  $R(\mathbf{w})$  is a regularization term,  $\lambda$  is a hyper-parameter which controls the strength of the regularization,  $\mathbf{o}_i = (0, \dots, 0, 1_c, 0, \dots, 0)$ .

It has been mentioned that the use of the weighted averaging significantly improves the DF and allows us to solve various machine learning problems by controlling the objective function for computing optimal weights [37, 38]. However, we need a more complex function of the class probability distributions sometimes in order to get superior results. This function can be implemented by means of neural networks which will be considered in the next section.

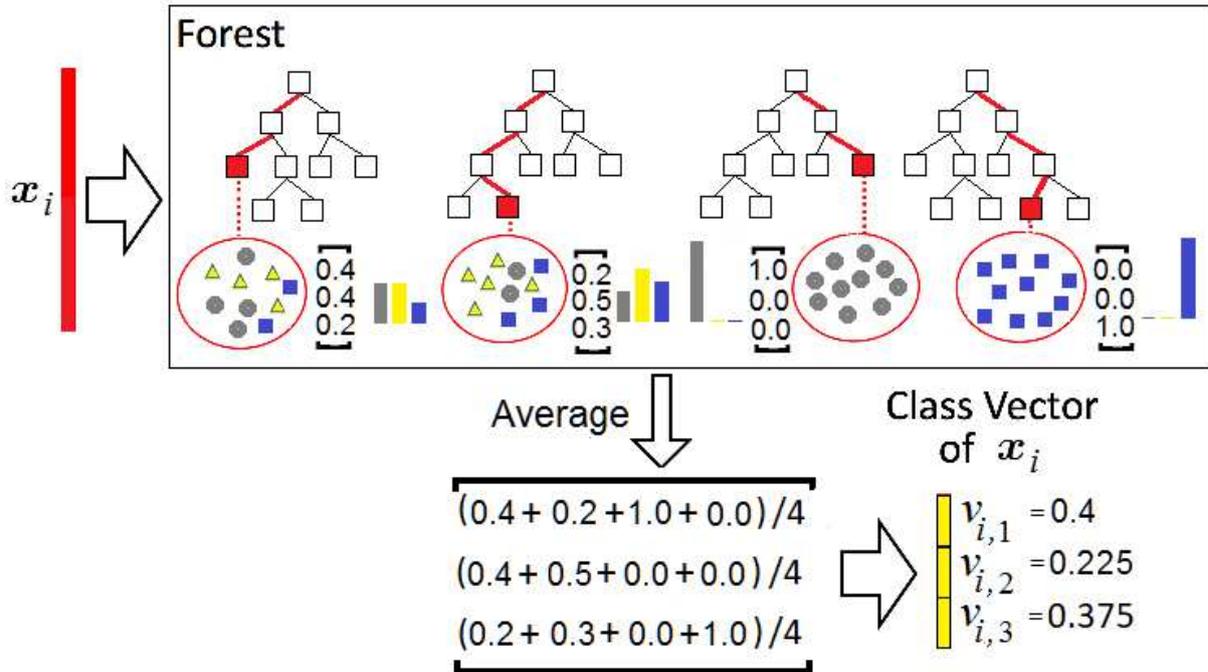


Figure 2: An illustration of the class vector generation by using average of the tree probability class vectors.

#### 4 Neural networks as a function of class probabilities

Let us return to the weighted averaging. The value  $v_{i,c}$  can be represented as a function  $f$  of probabilities  $\mathbf{p}_{i,c}$ , i.e.,  $v_{i,c} = f(\mathbf{p}_{i,c})$ . It is important to point out that the function  $f$  does not depend on the class  $c$ . At the same time, it is identical for all classes. Suppose now that the function  $f$  is not linear and is implemented by using the neural network. This implies that, for every class, we have to identically transform the vector  $\mathbf{p}_{i,c}$  in order to get the vector  $\mathbf{v}_i$  for every forest. It can be done by using  $C$  identical neural networks with shared parameters. The input of the  $c$ -th network is the vector  $\mathbf{p}_{i,c}$  of the length  $T$ . The output of the  $c$ -th network is expected to be 1 if the class label of the  $i$ -th instance coincides with the number of the network, i.e., if the condition  $y_i = c$  is valid, otherwise the output is expected to be 0. The networks are trained on the basis of sets of vectors  $\mathbf{p}_{i,c}$  obtained for every training example  $(\mathbf{x}_i, y_i)$ ,  $i = 1, \dots, n$ . The condition for training is that parameters of all networks have to be identical, i.e., the networks are implemented with shared parameters. This implies that that all networks are trained simultaneously.

Fig. 4 illustrates the use of identical neural networks with shared parameters for computing the class vectors. It can be seen from the picture that the input vector for the first neural network consists of first class probabilities of class probability distributions produced by all trees, i.e., it is the vector  $(0.4, 0.2, 1.0, 0.0)$ . The input vector for the second neural network consists of probabilities of the sec-

ond class, i.e., it is the vector  $(0.4, 0.5, 0, 0)$ . The same can be written for the third network input vector. In other words, the  $k$ -th network uses all probabilities of the  $k$ -th class. In the case of two classes, we have the standard Siamese neural network [6].

It should be noted that one network, say the last one, is superfluous because the  $C$ -th element of the vector  $\mathbf{v}_i$  can be obtained from its other elements under condition that the sum of all probabilities should be equal to 1. However, we use it in order to compensate a possible bias of probabilities.

A total algorithm of training the DF is given as Algorithm 1.

Having the trained NeuDF, we can make decision about the class of a new example  $\mathbf{x}$ . By using the trained decision trees and the neural networks, the vector  $\mathbf{x}$  is augmented at each level. Finally, we get the vector  $\mathbf{v}_i$  of augmented features after the  $Q$ -th level of the forest cascade corresponding to the original example  $\mathbf{x}$ . The example  $\mathbf{x}$  belongs to the class  $c$ , if the sum of the  $c$ -th elements of all vectors  $\mathbf{v}_i$  obtained for all RFs and all cascades (the total number of vectors is  $\sum_{q=1}^Q M_q$ ) is maximal.

The preliminary numerical experiments show that the proposed combination of the RFs and the neural networks may lead to overfitting. This is caused by a large number parameters of neural networks when the number of decision trees is also large because the number of trees defines the input vector for the neural networks. We have the following contradiction. On the one hand, we try to increase the number of trees in a RF in order to get better results. On the other hand, we have to use in this case a large neu-

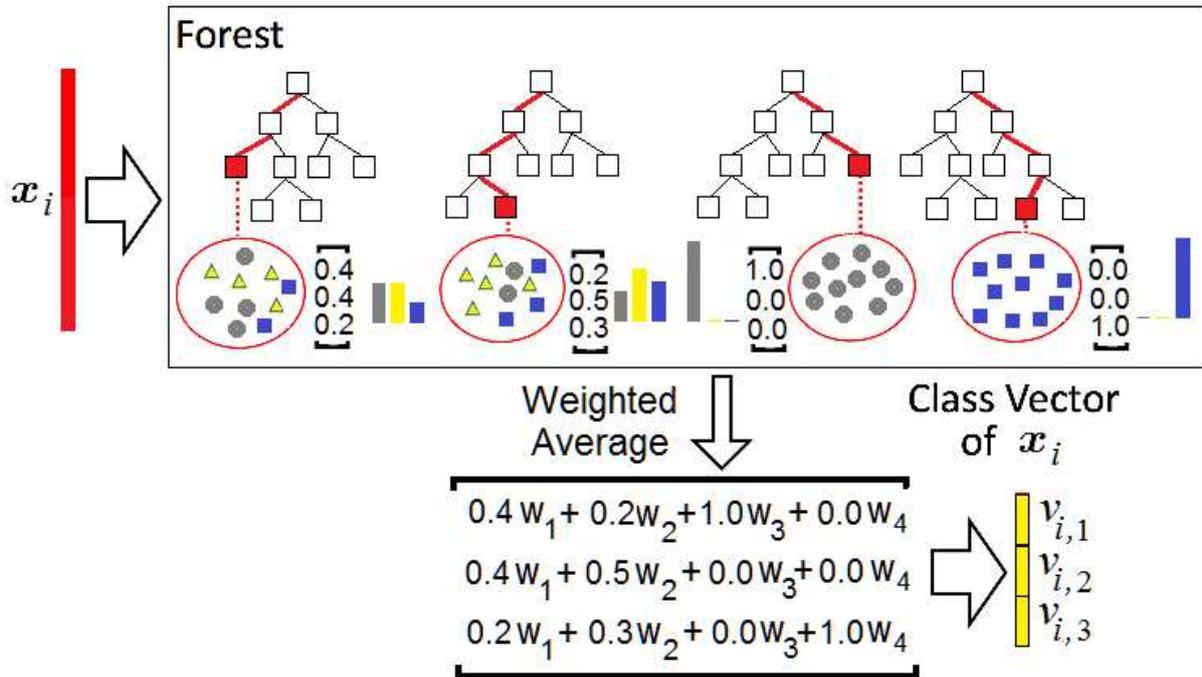


Figure 3: An illustration of the class vector generation taking into account the weights.

---

**Algorithm 1** A total algorithm for training the NeuDF

---

**Require:** Training set  $S = \{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$ ,  $\mathbf{x}_i \in \mathbb{R}^m$ ,  $y_i \in \{1, \dots, C\}$ ; number of levels  $Q$ ; number of forests at the  $q$ -th level  $M_q$

**Ensure:**  $w$  for every  $q = 1, \dots, Q$  and every  $k = 1, \dots, M_q$

- 1: **for**  $q = 1, q \leq Q$  **do**
  - 2:   **for**  $k = 1, k \leq M_q$  **do**
  - 3:     Train all trees from the  $k$ -th forest at the  $q$ -th level in accordance with the gcForest algorithm [45]
  - 4:     For every  $\mathbf{x}_i$ , compute  $C$  vectors of probabilities  $\mathbf{p}_{i,c}$ ,  $c = 1, \dots, C$
  - 5:     Train  $C$  neural networks from the  $k$ -th forest at the  $q$ -th level
  - 6:     For every  $\mathbf{x}_i$ , compute  $\mathbf{v}_i$  by using the trained neural networks for the  $k$ -th forest
  - 7:     Concatenation  $\mathbf{x}_i \leftarrow (\mathbf{x}_i, \mathbf{v}_i)$
  - 8:   **end for**
  - 9:   The concatenated vector  $\mathbf{x}_i$  is used for the next level
  - 10: **end for**
- 

ral network with many parameters (weights), which may lead to overfitting by a small training dataset. In order to overcome this difficulty, we proposed to use small neural networks with input vector of the dimensionality  $s$ . Here  $s$  is a tuning parameter. At that, all trees are united into groups such that there are  $s$  groups. The class probability distribution for every group is determined by averaging all class probability distributions in the group.

## 5 Numerical experiments

In order to illustrate NeuRF and NeuDF, we compare them with the gcForest. NeuDF has the same cascade structure as the standard gcForest described in [45]. Each level of the cascade structure consists of 10 RFs. In NeuDF, we do not use the Multi-Grained Scanning part. Three-fold cross-validation is used for the class vector generation. The number of cascade levels is 4.

NeuRF and NeuDF use a software in Python implementing the gcForest, which is available at <https://github.com/leopiney/deep-forest> to implement the procedure for computing optimal weights of trees and the corresponding class vectors. Accuracy measure  $A$  used in numerical experiments is the proportion of correctly classified cases on a sample of data. To evaluate the average accuracy, we perform a cross-validation with 100 repetitions, where in each run, we randomly select  $N$  training data and  $N_{\text{test}} = 3N/4$  test data.

The neural network in most numerical experiments consists of two hidden layers (total four layers). The number

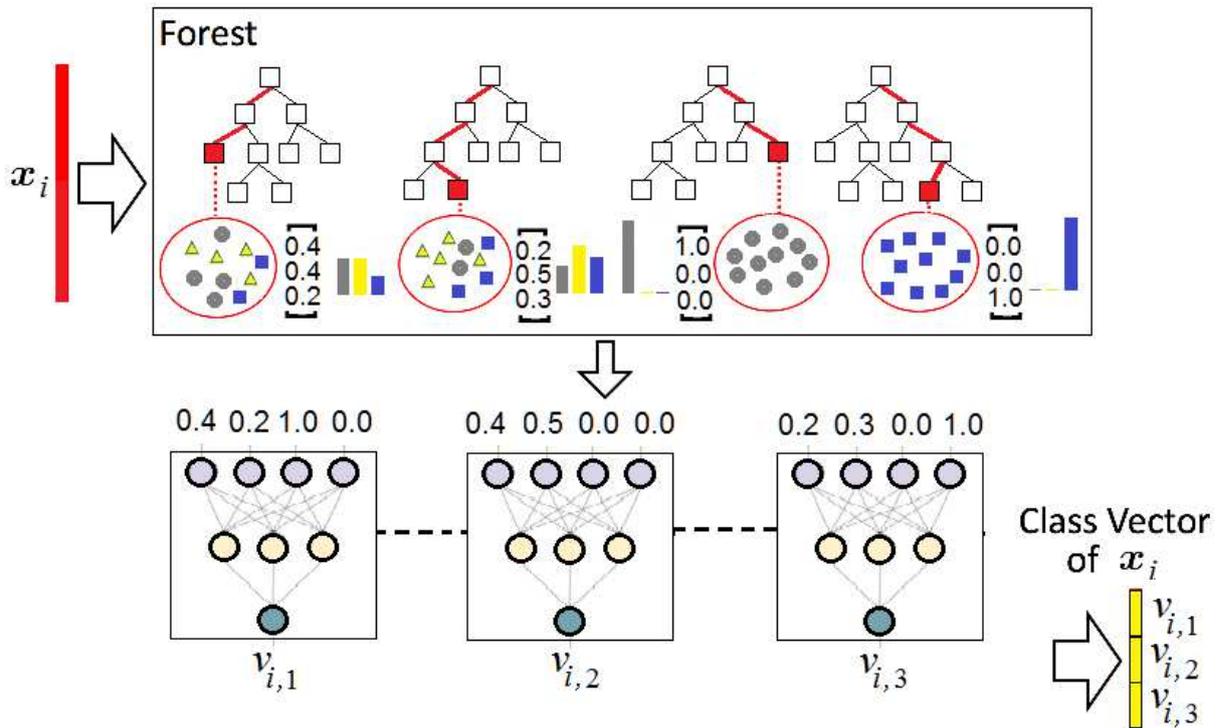


Figure 4: An illustration of the class vector generation by using  $C$  identical neural networks with shared parameters.

of neurons on the first hidden layer increases by 10% of the input layer. For example, if the input vector consists of 100 features then the first hidden layer contains 110 neurons. On the second layer, it decreases by 10% relative to the input layer, that is, consists of 90 neurons. However, we also investigate how the accuracy measures depend on the number of hidden layers in the neural network. The activation function is the sigmoid. The neural network is trained by using 50 epochs. The value of tuning parameter  $s$  is taken 4. Some numerical experiments illustrate the dependence of the classification accuracy on the parameter  $s$ . The number of decision trees in every RF is taken 1000. However, we also study how the number of trees impact the classification accuracy.

First, we compare NeuRF and NeuDF with the RF and gcForest, respectively, by using some public datasets from UCI Machine Learning Repository [26]. Table 1 is a brief introduction about these datasets, while more detailed information can be found from, respectively, the data resources. Table 1 shows the number of features  $m$  for the corresponding dataset, the number of examples  $n$  and the number of classes  $C$ . Different values for the regularization hyper-parameter  $\lambda$  have been tested, choosing those leading to the best results.

We also investigate the proposed models by using the well-known datasets: MNIST and CIFAR-10. The MNIST dataset is a commonly used large database of  $28 \times 28$  pixel handwritten digit images [24]. It has a training set of 60,000 examples, and a test set of 10,000 examples. The digits are size-normalized and cen-

tered in a fixed-size image. The dataset is available at <http://yann.lecun.com/exdb/mnist/>. The CIFAR-10 data set consists of  $32 \times 32$  color images drawn from 10 categories. It consists of 50,000 training and 10,000 test images each. It was collected by Krizhevsky et al. [22]. The data set is available at <https://www.cs.toronto.edu/~kriz/cifar.html>.

Numerical results of comparison of the RF and NeuRF are shown in Table 2, where the first column contains abbreviations of the tested data sets, the second column is the accuracy measure by using the RF, the third column contains the accuracy measures of NeuRF, and the fourth column represents the difference between the accuracy measures of NeuRF and the RF. It can be seen from Table 2 that the proposed NeuRF outperforms the RF for most considered data sets. However, we have to point out that this outperformance is not significant. In order to formally compare the proposed NeuRF with the RF, we apply the  $t$ -test which has been proposed and described by Demsar [12] for testing whether the average difference in the performance of two classifiers is significantly different from zero. Since we use the differences between accuracy measures of NeuRF with the RF (see Table 2), then we compare them with 0. The  $t$  statistics in this case is distributed according to the Student distribution with  $16 - 1$  degrees of freedom. The results of computing the  $t$  statistics for the difference are the p-value denoted as  $p$  and the 95% confidence interval for the mean 0.198, which are  $p = 0.036$  and  $[0.0139, 0.3823]$ , respectively. The  $t$ -test demonstrates the outperforming of NeuRF in comparison with the RF, but the p-value is very close to the bound (0.05) of accepting

Table 1: A brief introduction about data sets

Dataset	Abbreviation	$m$	$n$	$C$
Mammographic masses	MM	5	961	2
Haberman's Survival	HS	3	306	2
Seeds	Seeds	7	210	3
Ionosphere	Ion	34	351	2
Ecoli	Ecoli	8	336	8
Yeast	Yeast	8	1484	8
Parkinson	Park	23	351	2
Glass Identification	Glass	10	214	7
Indian Liver Patient Dataset	ILPD	10	583	2
Car Evaluation	Car	6	1728	4
Waveform Database Generator	Wave	40	5000	3
Soybean (Small)	Soyb	35	47	4
Wholesale Customer Region	WCR	8	440	3
Diabetic Retinopathy	Diab	20	1151	2
Mice Protein Expression	Mice	82	1080	8
Teaching Assistant Evaluation	TAE	5	151	3

Table 2: Comparison of RFs with modified RFs

Dataset	RF	NeuRF	Difference
MM	81.20	81.27	0.07
HS	73.02	73.24	0.22
Seeds	90.29	90.31	0.02
Ion	89.10	89.40	0.3
Ecoli	84.13	85.15	1.02
Yeast	58.11	58.45	0.34
Park	88.75	88.79	0.04
Glass	89.39	89.23	-0.16
ILPD	72.60	72.81	0.21
Car	88.97	89.17	0.2
Wave	84.89	84.32	-0.57
Soyb	85.10	85.46	0.36
WCR	74.83	74.96	0.13
Diab	71.19	71.20	0.01
Mice	94.20	94.86	0.66
TAE	53.17	53.49	0.32

Table 3: Comparison of the NeuDF with DF

Dataset	DF	NueDF	Difference
MM	82.90	83.85	0.95
HS	73.5	74.19	0.69
Seeds	91.2	92.31	1.11
Ion	90.3	91.05	0.75
Ecoli	88.23	89.30	1.07
Yeast	59	59.34	0.34
Park	89.7	90.07	0.37
Glass	90.1	91.31	1.21
ILPD	73.12	73.90	0.78
Car	89.5	90.34	0.84
Wave	85.01	85.91	0.9
Soyb	86.3	87.59	1.29
WCR	75.1	75.20	0.1
Diab	71.29	71.36	0.07
Mice	95.2	95.78	0.58
TAE	53.82	54.03	0.21

the null hypothesis, which means that the accuracy measures are not significantly different.

However, quite different results are obtained by comparing NeuDF and the DF. Numerical results of comparison of the DF and NeuDF are shown in Table 3. It can be seen from Table 3 that the proposed NeuDF outperforms the DF for all considered data sets. Moreover, the results of computing the  $t$  statistics for the differences between NeuDF and the DF (see Table 3) are the 95% confidence interval  $[0.495, 0.912]$  for the mean 0.704 with  $p = 0.000003$ . The  $t$ -test demonstrates the clear outperforming of NeuDF in comparison with the DF.

Let us formally compare also the RF and NeuDF as a extreme cases among the considered models models. By computing the  $t$  statistics for the differences between

NeuDF and the RF, we get the 95% confidence interval  $[1.047, 2.277]$  for the mean 1.662 with  $p = 0.000038$ . We see that NeuDF significantly outperforms the RF.

The same can be said about the MNIST and CIFAR datasets. The corresponding numerical results are shown in Table 4. One can see from Table 4 that NeuRF and NeuDF clearly outperform the RF and the DF, respectively.

Another question is how the accuracy measures of

Table 4: Comparison of the RFs and DFs with their modifications for MNIST and CIFAR data sets

Dataset	RF	NeuRF	DF	NeuDF
MNIST	96.04	96.44	98.4	99.20
CIFAR-10	93.90	94.32	94.89	95.44

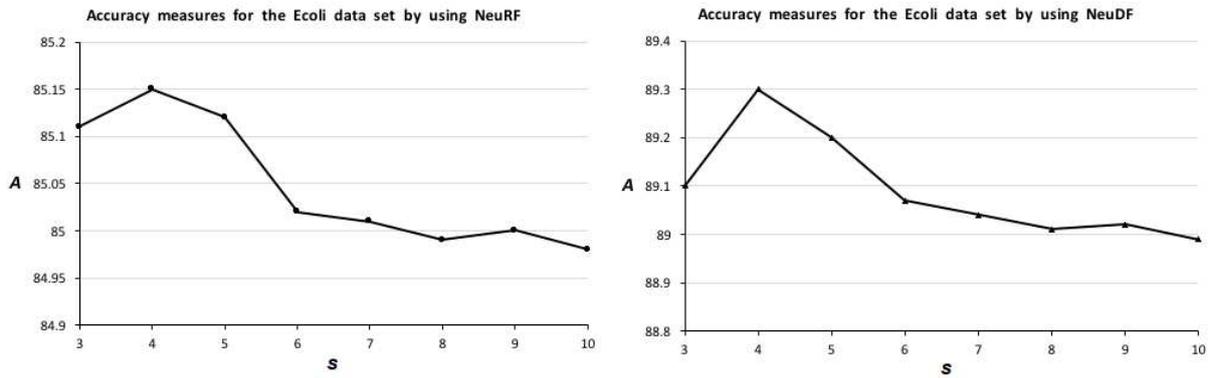


Figure 5: Accuracy measures as a function of the decision tree group numbers for the Ecoli dataset.

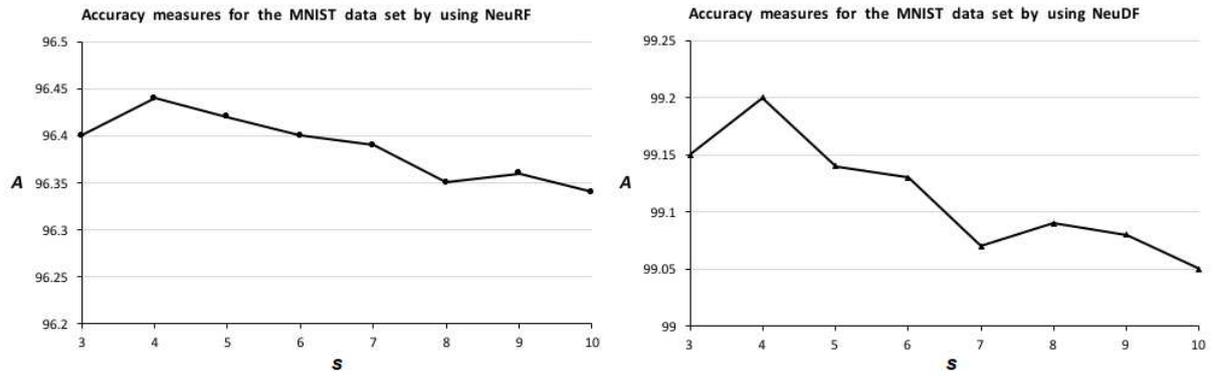


Figure 6: Accuracy measures as a function of the decision tree group numbers for the MNIST dataset.

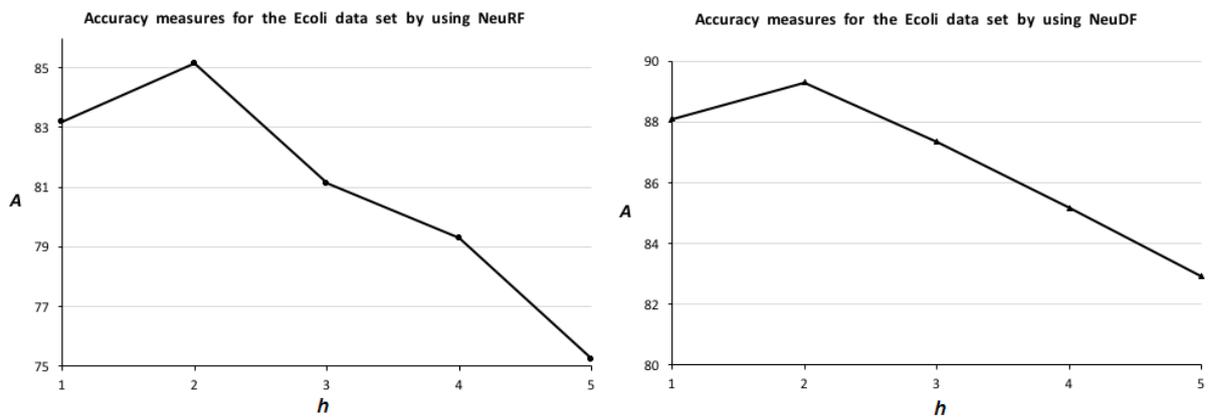


Figure 7: Accuracy measures as a function of the hidden layer numbers for the Ecoli dataset.

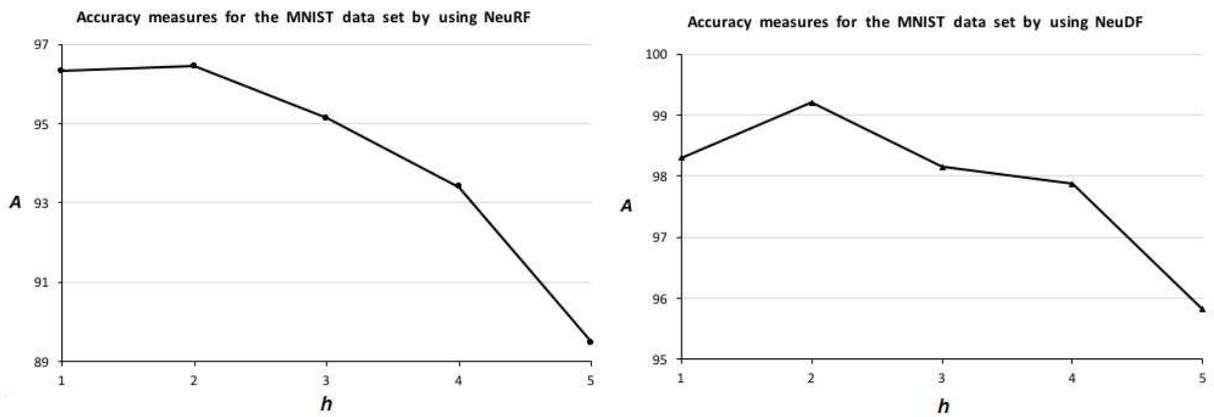


Figure 8: Accuracy measures as a function of the hidden layer numbers for the MNIST dataset.

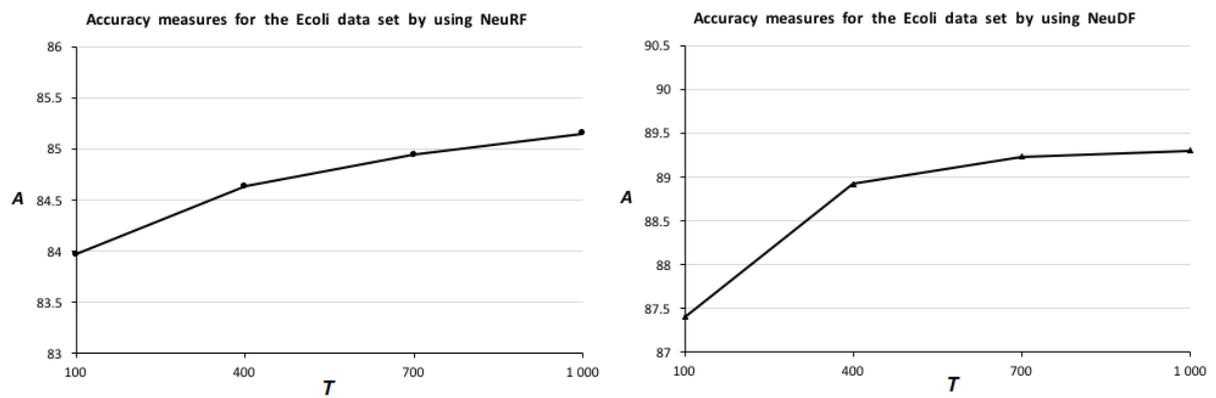


Figure 9: Accuracy measures as a function of the decision tree numbers in every RF for the Ecoli dataset.

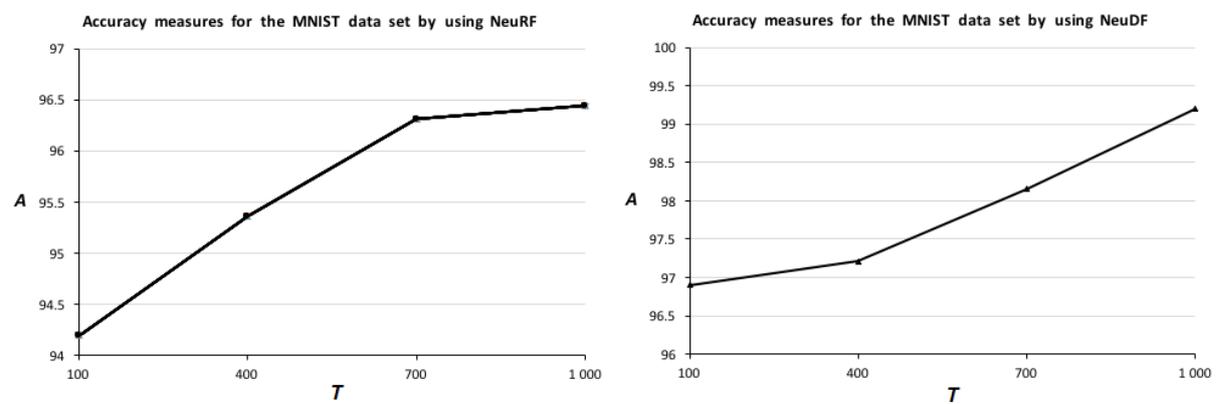


Figure 10: Accuracy measures as a function of the decision tree numbers in every RF for the MNIST dataset.

NeuRF and NeuDF depend on the decision tree group numbers  $s$ , i.e., on the tuning parameter  $s$ . Fig. 5 illustrates these dependences for the Ecoli dataset by using NeuRF (the left plot) and NeuDF (the right plot). It can be seen from the obtained results that there is an optimal value  $s$  which provides the largest accuracy. This value is 4, and it coincides for NeuRF as well as for NeuDF. The same results are obtained for the MNIST dataset (see Fig. 6). It is interesting to note that the optimal values of  $s$  coincide for the Ecoli and MNIST datasets. However, this is just a coincidence. If we perform the same numerical experiments, for example, with the Yeast dataset, then we get optimal value  $s = 6$ .

We also investigate how the number of hidden layers  $h$  in every neural network impacts on the accuracy measures. The corresponding curves are shown in Figs. 7-8. Here we again have an optimal value of  $h$ , which provides the largest accuracy. It is interesting to note that the increase of the hidden layers does not improve the results. Moreover, this increase makes the results worse. It can be explained by the overfitting effect when a lot of training parameters of the modified RF (weights of trees) are replaced by a lot of connection weights of the neural network. Finally, we investigate how the accuracy measures depend on the number  $T$  of decision trees in every RF. Figs. 9-10 clearly shows that the accuracy measures increase with  $T$ , but the computational complexity increases also in this case.

## 6 Conclusion

New classification models based on combination of the DF and the neural network have been presented in the paper. The main idea underlying these models is to improve RFs and the DF by combining the class probability distributions produced by decision trees for every training example by using a series of identical shallow neural networks with shared weights.

The proposed models have a number of advantages. First of all, we replace a simple rule for the class probability distribution combination (averaging) by a more complex function implemented by the neural network, which aims to minimize a classification loss function. Second, the neural network allows us to simply use various loss functions for computing the optimal RF class probability distributions. This leads to opportunity to solve tasks different from the standard classification, for example, transfer learning. Moreover, by applying the proposed models, we can modify the stacking algorithm used in the DF extending a set of the augmented features by some new functions of the tree class probability vectors. The investigation of new augmented features is a very interesting problem which can be viewed as a direction for further research.

It should be noted that the proposed models have not demonstrated a significant improvement when they were applied to a separate RF. A small increase of the accuracy

measures for many datasets in this case is compensated by additional computations because of the neural network training. However, numerical experiments have illustrated that the proposed combinations may be very effective for the DF because it forms the appropriate augmented features in the stacking algorithm. That is why we have considered modifications of RFs as well as the DF in the paper.

The neural networks in the proposed models are trained by using a training part of datasets. At the same time, a direction for further research is to change the neural network learning strategy. For example, they may learn by using testing data or a combination of training and testing data. The above changes may lead to outperforming results.

## Acknowledgement

This work is supported by the Russian Science Foundation under grant 18-11-00078.

## References

- [1] L. Bertinetto, J. Valmadre, J.F. Henriques, A. Vedaldi, and P.H.S. Torr. Fully-convolutional siamese networks for object tracking. *arXiv:1606.09549v2*, 14 Sep 2016.
- [2] G. Biau and E. Scornet. A random forest guided tour. *arXiv:1511.05741v1*, Nov 2015.
- [3] G. Biau, E. Scornet, and J. Welbl. Neural random forests. *arXiv:1604.07143v1*, Apr 2016.
- [4] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. <https://doi.org/10.1023/A:1018054314350>
- [5] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. <https://doi.org/10.1023/A:1010933404324>
- [6] J. Bromley, J.W. Bentz, L. Bottou, I. Guyon, Y. LeCun, C. Moore, E. Sackinger, and R. Shah. Signature verification using a siamese time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(4):737–744, 1993. <https://doi.org/10.1142/S0218001493000339>
- [7] T. Christensen C. Hettinger, B. Ehlert, J. Humpherys, T. Jarvis, and S. Wade. Forward thinking: Building and training neural networks one layer at a time. *arXiv:1706.02480v1*, Jun 2017.
- [8] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages

- 539–546. IEEE, 2005. <https://doi.org/10.1109/CVPR.2005.202>
- [9] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 7(2-3):81–227, 2011. <https://doi.org/10.1561/06000000035>
- [10] M.E.H. Daho, N. Settouti, M.E.A. Lazouni, and M.E.A. Chikh. Weighted vote for trees aggregation in random forest. In *2014 International Conference on Multimedia Computing and Systems (ICMCS)*, pages 438–443. IEEE, April 2014. <https://doi.org/10.1109/ICMCS.2014.6911187>
- [11] R.A. Dara, M.S. Kamel, and N. Wanas. Data dependency in multiple classifier systems. *Pattern Recognition*, 42(7):1260 – 1273, 2009. <https://doi.org/10.1016/j.patcog.2008.11.035>
- [12] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.
- [13] K. Fawagreh, M.M. Gaber, and E. Elyan. Random forests: from early developments to recent advancements. *Systems Science & Control Engineering*, 2(1):602–609, 2014. <https://doi.org/10.1080/21642583.2014.956265>
- [14] A.J. Ferreira and M.A.T. Figueiredo. Boosting algorithms: A review of methods, theory, and applications. In C. Zhang and Y. Ma, editors, *Ensemble Machine Learning: Methods and Applications*, pages 35–85. Springer, New York, 2012. [https://doi.org/10.1007/978-1-4419-9326-7\\_2](https://doi.org/10.1007/978-1-4419-9326-7_2)
- [15] R. Genuer, J.-M. Poggi, C. Tuleau-Malot, and N. Villa-Vialaneix. Random forests for big data. *Big Data Research*, 9:28–46, 2017. <https://doi.org/10.1016/j.bdr.2017.07.003>
- [16] M. Hibino, A. Kimura, T. Yamashita, Y. Yamauchi, and H. Fujiyoshi. Denoising random forests. arXiv:1710.11004v1, Oct 2017.
- [17] J. Hu, J. Lu, and Y.-P. Tan. Discriminative deep metric learning for face verification in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1875–1882. IEEE, 2014. <https://doi.org/10.1109/CVPR.2014.242>
- [18] Y. Ioannou, D. Robertson, D. Zikic, P. Kotschieder, J. Shotton, M. Brown, and A. Criminisi. Decision forests, convolutional networks and the models in-between. arXiv:1603.01250v1, Mar 2016.
- [19] A. Jurek, Y. Bi, S. Wu, and C. Nugent. A survey of commonly used ensemble-based classification techniques. *The Knowledge Engineering Review*, 29(5):551–581, 2014. <https://doi.org/10.1017/S0269888913000155>
- [20] H. Kim, H. Kim, H. Moon, and H. Ahn. A weight-adjusted voting algorithm for ensemble of classifiers. *Journal of the Korean Statistical Society*, 40(4):437–449, 2011. <https://doi.org/10.1016/j.jkss.2011.03.002>
- [21] P. Kotschieder, M. Fiterau, A. Criminisi, and S.R. Bulu. Deep neural decision forests. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1467–1475, 2015. <https://doi.org/10.1109/ICCV.2015.172>
- [22] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical Report 1, Computer Science Department, University of Toronto, 2009.
- [23] L.I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, New Jersey, 2004.
- [24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. <https://doi.org/10.1109/5.726791>
- [25] H. B. Li, W. Wang, H. W. Ding, and J. Dong. Trees weighting random forest method for classifying high-dimensional noisy data. In *2010 IEEE 7th International Conference on E-Business Engineering*, pages 160–163. IEEE, Nov 2010. <https://doi.org/10.1109/ICEBE.2010.99>
- [26] M. Lichman. UCI machine learning repository, 2013.
- [27] G. Louppe. Understanding random forests: From theory to practice. arXiv:1407.7502v3, June 2015.
- [28] D. Maji, A. Santara, S. Ghosh, D. Sheet, and P. Mitra. Deep neural network and random forest hybrid architecture for learning to detect retinal vessels in fundus images. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 3029–3032. IEEE, Aug 2015. <https://doi.org/10.1109/EMBC.2015.7319030>
- [29] K. Miller, C. Hettinger, J. Humpherys, T. Jarvis, and D. Kartchner. Forward thinking: Building deep random forests. arXiv:1705.07366, 20 May 2017.
- [30] R. Polikar. Ensemble learning. In C. Zhang and Y. Ma, editors, *Ensemble Machine Learning: Methods and Applications*, pages 1–34. Springer,

- New York, 2012. [https://doi.org/10.1007/978-1-4419-9326-7\\_1](https://doi.org/10.1007/978-1-4419-9326-7_1)
- [31] Y. Ren, L. Zhang, and P. N. Suganthan. Ensemble classification and regression-recent developments, applications and future directions [review article]. *IEEE Computational Intelligence Magazine*, 11(1):41–53, 2016. <https://doi.org/10.1109/MCI.2015.2471235>
- [32] L. Rokach. Ensemble-based classifiers. *Artificial Intelligence Review*, 33(1-2):1–39, 2010. <https://doi.org/10.1007/s10462-009-9124-7>
- [33] L. Rokach. Decision forest: Twenty years of research. *Information Fusion*, 27:111–125, 2016. <https://doi.org/10.1016/j.inffus.2015.06.005>
- [34] C.A. Ronao and S.-B. Cho. Random forests with weighted voting for anomalous query access detection in relational databases. In *Artificial Intelligence and Soft Computing. ICAISC 2015*, volume 9120 of *Lecture Notes in Computer Science*, pages 36–48, Cham, 2015. Springer. [https://doi.org/10.1007/978-3-319-19369-4\\_4](https://doi.org/10.1007/978-3-319-19369-4_4)
- [35] W. Shen, Y. Guo, Y. Wang, K. Zhao, B. Wang, and A. Yuille. Deep regression forests for age estimation. arXiv:1712.07195v1, Dec 2017.
- [36] W. Shen, K. Zhao, Y. Guo, and A. Yuille. Label distribution learning forests. arXiv:1702.06086v4, Oct 2017.
- [37] L.V. Utkin and M.A. Ryabinin. Discriminative metric learning with deep forest. arXiv:1705.09620v1, May 2017.
- [38] L.V. Utkin and M.A. Ryabinin. A deep forest for transductive transfer learning by using a consensus measure. In A. Filchenkov, L. Pivovarova, and J. Zizka, editors, *Artificial Intelligence and Natural Language. AINL 2017*, volume 789 of *Communications in Computer and Information Science*, pages 194–208. Springer, Cham, 2018. [https://doi.org/10.1007/978-3-319-71746-3\\_17](https://doi.org/10.1007/978-3-319-71746-3_17)
- [39] L.V. Utkin and M.A. Ryabinin. A Siamese deep forest. *Knowledge-Based Systems*, 139:13–22, 2018. <https://doi.org/10.1016/j.knosys.2017.10.006>
- [40] S. Wang, C. Aggarwal, and H. Liu. Using a random forest to inspire a neural network and improving on it. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 1–9. Society for Industrial and Applied Mathematics, Jun 2017. <https://doi.org/10.1137/1.9781611974973.1>
- [41] D.H. Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992. [https://doi.org/10.1016/S0893-6080\(05\)80023-1](https://doi.org/10.1016/S0893-6080(05)80023-1)
- [42] M. Wozniak, M. Grana, and E. Corchado. A survey of multiple classifier systems as hybrid systems. *Information Fusion*, pages 3–17, 2014. <https://doi.org/10.1016/j.inffus.2013.04.006>
- [43] P. Yang, E.H. Yang, B.B. Zhou, and A.Y. Zomaya. A review of ensemble methods in bioinformatics. *Current Bioinformatics*, 5(4):296–308, 2010. <https://doi.org/10.2174/157489310794072508>
- [44] Z.-H. Zhou. *Ensemble Methods: Foundations and Algorithms*. CRC Press, Boca Raton, 2012.
- [45] Z.-H. Zhou and J. Feng. Deep forest: Towards an alternative to deep neural networks. arXiv:1702.08835v2, May 2017.
- [46] J. Zhu, Y. Shan, J.C. Mao, D. Yu, H. Rahmani, and Y. Zhang. Deep embedding forest: Forest-based serving with deep embedding features. arXiv:1703.05291v1, Mar 2017.



# Creation of Facial Composites from User Selections Using Image Gradients

Rubén García-Zurdo

Universidad Complutense de Madrid, Facultad de Psicología, Madrid, Spain

The Open University, School of Physical Sciences, Milton Keynes, UK

E-mail: rubengarciazurdo@gmail.com

**Keywords:** facial composites, human-computer interfacing, image gradient, poisson editing

**Received:** May 27, 2018

*Evolutionary facial composites are created using interactive genetic algorithms based on user selections. This approach is grounded in perceptive studies, and is superior to feature-based systems. A method is presented for creating facial composites in which faces are encoded with shape information, the coordinates of a predefined landmark points, and the image gradient, which represents face information more precisely than image luminance. The new method is accompanied by a Poisson integration process that presents the user with candidate faces. Two user tests, one using composite creators and the other external evaluators, show that the new method produces higher rated composites that are better recognised.*

*Povzetek: Opisana je metoda generiranja slik za prepoznavo na osnovi interaktivnega genetskega algoritma.*

## 1 Introduction

The goal of facial compositing systems is to create a face image of a target identity from a person's memory so it can be recognised by other people. There are two categories of computerised facial composite systems: in feature-based systems, such as E-FIT [1] and PRO-fit [2], the operator selects features as the eyes, nose and mouth and arranges them on a template to create a face from its parts, while in holistic or evolutionary facial compositing, the operator evolves a whole face by 'breeding' selections from an array of face images, via a process of selection by recognition [3]. Systems in the latter category include EFIT-V [4], ID [4], INIH [6] and EvoFIT [7]. Many of these systems lack a formal user test that can verify their real utility, and identification of individuals from facial composites remains generally poor, meaning that searches for new approaches are justified.

EvoFIT is the system that has been most extensively studied. It produces composites that are identified correctly 30% of the time by people who are familiar with the target identities [8]. This can rise to 45% using more recent strategies for composition [9]. Humberside police used EvoFIT in 35 criminal investigations, and it led to arrests in 60% of cases [10].

Facial compositing research has also produced or confirmed several results that are relevant to face perception: the importance of the internal features of faces over external features [11], [12], [13], [14], the relevance of using configural information [15] and holistic dimensions to describe faces, such as masculinity [16], [17], and the unimportance of colour for face recognition and compositing [18], [19], [20].

Evolutionary face compositing uses interactive genetic algorithms in which the operator selects a number of candidates in an iterative process. These algorithms use an evolutionary mechanism where face representations evolve through crossing (i.e. a mixing of genetic code from selected representations or parents) and random mutation occurring with a predefined low probability [3]. The human operator selects candidates from a gallery, and this selection acts as a fitness function to drive the system to converge to a final composite image resembling the remembered face.

The genetic code or representation of faces is a vector of principal component analysis (PCA) coefficients. PCA represents each face as a coefficient vector corresponding to the weights of a linear combination of elementary faces, called eigenfaces, which are obtained from a sample of images. Each eigenface possesses an associated eigenvalue indicating the amount of variance of the sample that is explained by it. Eigenfaces are usually ordered according to their eigenvalues in such a way that the first eigenfaces contribute more to explaining the observed variance in a sample of images than the remaining eigenfaces. Eigenfaces may be obtained by applying PCA [20] or singular value decomposition (SVD) to the normalised covariance matrix of a sample of images.

However, it is first necessary to align the face images. Although Procrustes analysis can achieve this optimally based on a set of facial landmark points, and can yield the necessary translation, rotation and scaling of shapes to get the best possible alignment, perfect alignment between faces is not usually possible because

each face has a unique shape. This problem is solved with a shape normalisation technique in which images are warped to a reference shape template so they become shape-free, and PCA is performed on the shape-free images [21]. The shape information of individual faces, represented as the x-y coordinates of the landmark points, is used to perform a second PCA to build a eigenshape representation. Each face is thus represented by a pair of texture and shape vectors of PCA coefficients.

Since the introduction of evolutionary facial compositing two decades ago [3], no new representations have been suggested in the literature, with the exception of a combined shape-texture PCA [4], and a user test that would measure the benefit from this approach is missing. Research into new kinds of face representations seems justified, as this may help with the important problem of the limited expressive power of eigenfaces to produce new faces that are not included in the sample as a linear combination of eigenfaces [22]. Face shape and texture are also independent cues for facial recognition [23], [24], [25] and it is therefore hypothesised that the specific method used to render texture in facial composites may have a significant impact on recognition.

Image gradient is introduced here as an alternative representation to facial texture. Image gradient is a differential transformation that represents the direction and magnitude of the maximum intensity change at each pixel by calculating the differences between adjacent pixels in the x and y directions [26]. It can be conceived as a representation of the derivative of a 2D function (i.e. the image) that produces peak responses in places where there is a sudden change of intensity (i.e. the edges). It was proposed as a basic mechanism in early visual processing, and edge detection algorithms have been developed based on this approach [27].

Image gradient represents the underlying structure of the elements in the image better than intensity, and so constitutes a more precise representation that is less affected by illumination patterns. This is illustrated in Figure 1, where the eigenvalues (or amount of associated variance) of the gradient of the facial images used here are shown versus the eigenvalues of components computed based on intensity. The gradient eigenvalues are more uniformly distributed than the intensity ones, which show an initial peak and then a sharp decrease. This peak corresponds to coarse luminance variations in the images [22] and is attenuated in the gradient representation, since the gradient only encodes the differences between adjacent pixels and not their absolute values.

The use of a gradient representation of the facial texture means that a gradient integration technique is necessary to present the corresponding intensity values to the participants. This integration problem is known as Poisson's equation, and is usually solved by setting conditions on the values taken at the area boundary and using an iterative solving method [28]. A major application of Poisson editing is to paste elements into images in a seamless way.

In the present implementation of the system, a constant value at the external edges of the face area is used as a boundary condition. Although this may seem a simplistic condition, it is sufficient to produce realistic images from its gradient. Figure 2 shows that a constant boundary condition is able to recover a individual face from its gradient, since most of the important information seems to be stored in the gradient rather than in the individual pixel values. Even small-range random values at the boundary are sufficient to recover the individual faces.

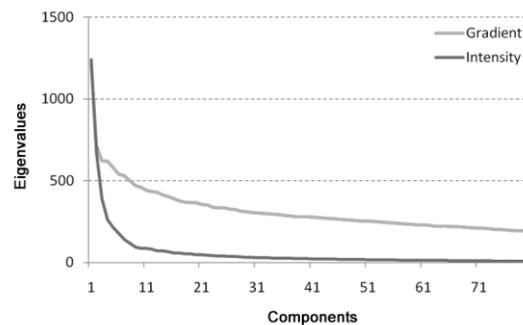


Figure 1: Variance of gradient and intensity PCA components.

The goal of this work is to describe an evolutionary system using the image gradient as a representation of texture and to compare the recognisability and likeness of the resulting composites with those produced using the standard intensity representation of face texture. An initial version of the system with some preliminary results was presented in [29]. Formal mathematical and implementation details are introduced in the appendix.

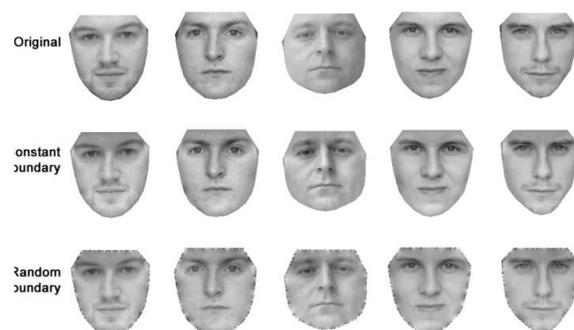


Figure 2: Intensity reconstruction from gradient using constant and random boundary values.

## 2 Method

The method is illustrated in Figure 3. Sixty-two pictures from the Glasgow unfamiliar face database [30] and 24 pictures from the Utrecht ECVF face database ([http://pics.stir.ac.uk/2D\\_face\\_sets.htm](http://pics.stir.ac.uk/2D_face_sets.htm)) were used as reference faces. This gave a total of 86 pictures of Caucasian males, who were mostly in their twenties in the Glasgow sample and in their thirties in the Utrecht sample. Each image shows a frontal view of a face under approximately frontal illumination. Sixty-eight facial landmarks were automatically located on each picture

using a robust state-of-the-art method based on machine learning [31]. Images were converted to grey-scale and warped to a reference shape using the thin plate spline technique. The shape, intensity and gradient PCAs were computed and the resulting components were used in the following genetic algorithm.

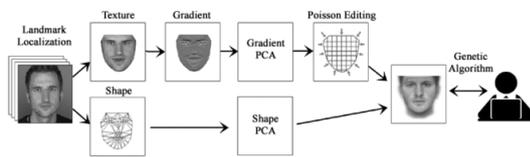


Figure 3: Evolutionary facial compositing overview.

### Algorithm

An interactive genetic algorithm is used with the aim of generating a facial image; in this approach, the human operator selects two candidates or parents from a gallery of six images in a 2x3 array. Each face is represented as two vectors, one containing shape coefficients (size 40) and the other texture coefficients (size 80).

- i. Random initialisation: Randomly select values from a uniform distribution of one standard deviation around each PCA component
- ii. Repeat for a number of generations:
  - a. Operator selects two parents
  - b. Breed a new generation by crossing parent vectors and adding random mutations for both shape and texture
  - c. Render candidate gallery for next generation
- iii. Keep selected final image in last generation as the final composite

## 3 Construction test

### Participants

Twenty students (15 women, five men) acted as constructor participants to build the face composites ( $M_{\text{age}} = 19.9$ ,  $SD = 1.48$  years). They took part in the experiment as an educational exercise in groups of five.

### Design and procedure

Participants received instructions to construct the face of six well-known male celebrities. These were: David Beckham (DB), George Clooney (GC), Nicolas Cage (NC), Robert De Niro (RN), Tom Cruise (TC) and Tom Hanks (TH). A photo-array of the celebrities was presented briefly to refresh their memory and confirm that all participants were familiar with the targets and their names. They received verbal instruction and hands-on training on how to select the two images most similar to the target identity in order of preference, by clicking the mouse. Participants could erase their selection at any time in order to change it, before proceeding to the next generation by pressing a "Continue" button.

For each generation, six images were shown in a 3x2 array in the centre of the screen. Each participant constructed a total of 12 composites, one for each of the six targets using two levels of representation (gradient and intensity). The order of construction of the 12

composites was varied randomly for each participant. After constructing the composites, participants were asked to rate the likeness of their own composites to the target identity on a scale of 1-10, where 1 means "absolutely dissimilar" and 10 "totally similar". In this case, composites were presented individually on the screen, with the target's name at the top, and the response was given by clicking a number with the mouse. Participants were also asked to rate each target identity in terms of distinctiveness on a scale of 1-10, where 1 means "not distinctive at all" and 10 "maximally distinctive". This time, only the name of each target was shown, so that participants based their response on their own internal representation. Distinctiveness was defined to them as "the degree to which a face would stand out from the rest of the faces in a crowd". The whole procedure took between 50 and 70 minutes for all participants. A one minute rest was allowed after finishing the creation of each composite.

### Results

Figure 4 shows examples of the final composites from a participant using gradient and intensity representations.

A within-subject two-way ANOVA was performed for likeness ratings made by constructor participants between Representation (Gradient, Intensity) and Target (DB, GC, NC, RN, TC, TH). A significant effect was obtained for Representation [ $F(1,19) = 51.33$ ,  $p < .05$ ,  $\eta^2 = .281$ ] in the comparison between gradient ( $M = 5.51$ ,  $SE = 0.32$ ) and intensity ( $M = 4.6$ ,  $SE = 0.23$ ), following the Greenhouse-Geisser correction. A similarly significant effect for target was obtained [ $F(5,95) = 3.23$ ,  $p < .05$ ,  $\eta^2 = .148$ ] in the comparison between target identities ( $M_{DB} = 6.17$ ,  $SE_{DB} = 0.34$ ;  $M_{GC} = 4.87$ ,  $SE_{GC} = 0.4$ ;  $M_{NC} = 4.45$ ,  $SE_{NC} = 0.36$ ;  $M_{RN} = 4.42$ ,  $SE_{RN} = 0.46$ ;  $M_{TC} = 5.47$ ,  $SE_{TC} = 0.45$ ;  $M_{TH} = 5.92$ ,  $SE_{TH} = 0.4$ ) with assumed sphericity following the Mauchly test. Multiple comparison tests revealed that differences existed between targets DB and NC [ $p < .001$ ] and DB and RN [ $p < .05$ ]. 41.7% of the gradient images received a rating equal to or greater than seven, while only 18.3% of the intensity images received similar ratings. No significant interaction was evident for the interaction of representation x target.

Additionally, a within-subject one-way ANOVA was performed to study the possible differences in target identity distinctiveness, which showed no significant difference. Separate correlation analyses were performed for the gradient and intensity representations for the individual distinctiveness ratings given by constructor participants and their corresponding likeness ratings. A non-significant correlation existed for the gradient representation [ $\rho = .13$ ,  $p = .163$ ] but a significant correlation existed for the intensity representation [ $\rho = .2$ ,  $p = .030$ ]. A linear regression analysis was then performed for the distinctiveness and likeness ratings for intensity representations, which proved to be significant [ $F(1,118) = 4.83$ ,  $p < 0.05$ ]. The corresponding scatter plot for distinctiveness and likeness and the linear model are shown in Figure 5.

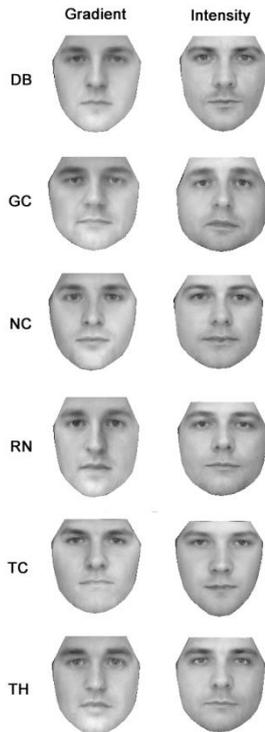


Figure 4: Final composites created with gradient and intensity representations.

**Discussion**

The composite constructors perceived a higher likeness between their own gradient-based composites and the target identity. Some identity composites tended to generate a higher likeness rating, and it is hypothesised that this was due to the facial distinctiveness of the target. Although we could not prove a significant difference by identity from the collected distinctiveness ratings, two separate correlation analyses of likeness and distinctiveness for the gradient and intensity-based composites showed a significant correlation only for intensity-based composites. This suggests that intensity-based composites are less able to capture the distinctiveness of some faces. This problem is somewhat minimised in gradient-based composites.

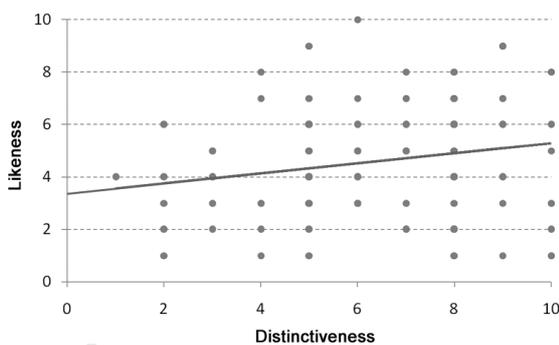


Figure 5: Scatter plot and linear model for likeness and distinctiveness ratings given by constructors for intensity-based composites.

**4 External evaluator test**

**Stimuli and material**

The 240 composite images built by the 20 constructor participants were used.

**Participants**

Forty psychology students (33 women, seven men) took part in the experiment as an educational exercise ( $M_{age} = 18.9, SD = 1.11$  years). They worked in small groups of five.

**Design and procedure**

Each participant performed two tasks (naming and likeness rating) using the composite images from four constructors. After briefly showing the participants the photo-array of celebrities, to confirm that they were all familiar with them and their names, a name-sorting task was used to measure composite recognition. The composites of the 20 constructors were partitioned into five blocks containing the resulting images of four constructors, corresponding to eight trials of each task (four at the gradient level of representation, and four at the intensity level).

In the naming task, each participant was asked to establish a correspondence between each of the six images presented, which were created by a constructor at a given representation level (gradient, intensity), and a target name. Images were presented in a 2x3 array with a clickable list of target names in alphabetical order underneath each image. The image order was varied randomly by trial, and representation-level blocks were varied randomly by participant. In the likeness rating task, the same composites were presented to each participant in random order. The presentation and response procedures were similar to those used by the constructor participants. The overall procedure took between 15 and 20 minutes for all participants.

**Results**

Two mixed ANOVA analyses with two between-subject factors (constructor and block) and one within-subject factor (representation) were performed on the percentage of correct naming and likeness ratings. The constructor and block were included as factors to account for any possible effect of the constructors' ability and specific block selection, meaning that their control acts as a measure of the quality of any difference found.

A significant difference was found between the likeness ratings for gradient ( $M = 3.88, SE = 0.12$ ) and intensity ( $M = 3.68, SE = 0.1$ ), following the Greenhouse-Geisser correction, although the effect size was small [ $F(1,140) = 4.08, p < .05, \eta^2 = .028$ ]. A significant difference was also found between correct namings for gradient ( $M = 21.04, SE = 1.4$ ) and intensity ( $M = 16.56, SE = 1.35$ ), with a somewhat greater effect size [ $F(1,140) = 6.09, p < .015, \eta^2 = .042$ ], following the Greenhouse-Geisser correction. No effects from the constructor, block or interaction between factors was detected for either likeness or naming.

## Discussion

A medium/small advantage in correct naming by external peers for gradient-based composites was found for the sample. We observed a trend of better recognition of the composites constructed using the gradient representation rather than the traditional intensity representation. It is therefore possible to hypothesise that since image gradient is a more invariant characteristic of the elements in an image, it should also represent facial features better than intensity.

We also observed a gradient advantage for likeness ratings given by external peers, although the effect size was smaller than for the constructor participants. As a proxy for naming, the likeness ratings do not always follow the same pattern of effect. There are two possible explanations for this discrepancy: either differences in rating criteria between participants, or differences in the exposure time and familiarity with similar composites between the constructors and external peers.

## 5 General discussion

Image gradient, an alternative method of representation to image intensity for evolutionary face compositing, was introduced, and its impact on the recognition and likeness ratings of composites was studied. The results indicate a benefit in terms of recognition for the gradient-based composites in our sample. Gradient-based composites are at least as good as those using the standard texture representation. It is conjectured that a benefit may arise from a better representation of facial features by gradient than by intensity. Facial PCA is a powerful tool for analysing facial data [3], but its ability to express new faces as a linear combination of components may be somewhat restricted. Eigenfaces were created for automatic face recognition (a discriminative task), and their ability to express new faces not present in the initial face database (a generative task) may be limited. In this work, a strategy has been followed that consisted of studying a different facial representation on which to perform evolution, in order to increase the representativeness of facial features and thus the accuracy and recognisability of facial composites. The variance associated with gradient components is distributed more uniformly than that associated with intensity components. This implies that during the random mutation stage of composite evolution, the range from which a value is selected is more homogeneous between components and the weight of components is more similar for gradient-based composites.

In previous research [13], a benefit was identified in terms of recognition using a sketch representation, which was presumably caused by a simplification of the facial texture that presented participants with a less demanding

situation. A sketch representation may be beneficial since less shading is involved, which results in less inaccurate information overall. This sketch model was computed for the EvoFIT face set in a preprocessing step, before applying PCA. A similar beneficial effect seems to be arising here from the use of facial image gradient.

As an additional test, automatic evolution of the system was performed for the same target identities as in the user test. The fitness function used was a correlation with an image of the target identity. The results were compared for the three kinds of texture representation, i.e. the intensity, the gradient-preprocessed intensity, in which the sample images were reconstructed from their gradient before PCA, and the gradient. The results shown in Figure 6 offer a visual comparison of their quality.

The evolutionary parameters used here (the numbers of shape and texture components, samples per generation, elitism, mutation and combination rates) were selected based on previous research on intensity representation. Further studies should be carried out to establish their optimal values for gradient representation. Given the huge amount of research on evolutionary facial composites, it should be noted that an ultimate conclusion on the superiority of a new face representation cannot be established from a single work, and extensive research comparing different situations should be conducted.

An improvement was made to our system after the formal experiments were carried out. The number of images presented to participants at each generation was initially six, since the time taken to perform gradient integration in the first implementation of the system (about three seconds per image) persuaded us not to use a greater number of images. This issue was solved in a new version of the system, where a 70% reduction in the time required for gradient integration now allows for the use of greater numbers of images and generations. New features have also been added to the system, such as another set of boundary conditions and the ability to add external features and depth to the resulting composites using optical flow methods. A previous study has at least explored the use of image gradient for facial compositing [32], although this was done from a featural point of view and used gradient integration to stitch fragments from different faces together. Another interesting venue is the exploratory use of deep learning generative adversarial networks for image generation [33] which could theoretically increase the generative power of compositing systems.

It is our conclusion that research on new approaches to face representation could improve the results of evolutionary facial compositing. The present system is available on request to face researchers as a Windows application, with no installation required.

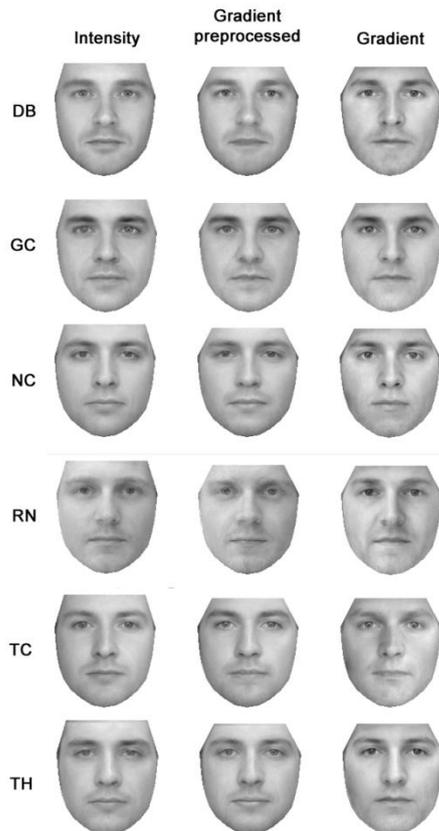


Figure 6: Results of automatic evolution using intensity, gradient-preprocessed and gradient representations.

## 6 References

- [1] Davies, G., van der Willik, P., & Morrison, L. J. (2000). Facial composite production: A comparison of mechanical and computer-driven systems. *Journal of Applied Psychology*, 85(1), 119. <https://doi.org/10.1037/0021-9010.85.1.119>
- [2] Frowd, C. D., McQuiston-Surrett, D., Anandaciva, S., Ireland, C. G., & Hancock, P. J. (2007). An evaluation of US systems for facial composite production. *Ergonomics*, 50(12), 1987-1998. <https://doi.org/10.1080/00140130701523611>
- [3] Hancock, P. J. (2000). Evolving faces from principal components. *Behavior Research Methods, Instruments, & Computers*, 32(2), 327-333. <https://doi.org/10.3758/bf03207802>
- [4] Solomon, C. J., Gibson, S. J., & Mist, J. J. (2013). Interactive evolutionary generation of facial composites for locating suspects in criminal investigations. *Applied Soft Computing*, 13(7), 3298-3306. <https://doi.org/10.1016/j.asoc.2013.02.010>
- [5] Tredoux, C., Nunez, D., Oxtoby, O., & Prag, B. (2006). An evaluation of ID: An eigenface based construction system. *South African Computer Journal*, 37, 90-97.
- [6] Kurt, B., Etaner-Uyar, A. S., Akbal, T., Demir, N., Kanlikilicer, A. E., Kus, M. C., & Ulu, F. H. (2006). Active appearance model-based facial composite generation with interactive nature-inspired heuristics. *International Workshop on Multimedia Content Representation, Classification and Security*, 2006. pp. 183-190. [https://doi.org/10.1007/11848035\\_26](https://doi.org/10.1007/11848035_26)
- [7] Frowd, C. D., Hancock, P. J., & Carson, D. (2004). EvoFIT: A holistic, evolutionary facial imaging technique for creating composites. *ACM Transactions on Applied Perception*, 1(1), 19-39. <https://doi.org/10.1145/1008722.1008725>
- [8] Frowd, C. D., Pitchford, M., Bruce, V., Jackson, S., Hepton, G., Greenall, M., ... & Hancock, P. J. (2011). The psychology of face construction: Giving evolution a helping hand. *Applied Cognitive Psychology*, 25(2), 195-203. <https://doi.org/10.1002/acp.1662>
- [9] Frowd, C. D., Skelton, F., Atherton, C., Pitchford, M., Hepton, G., Holden, L., ... & Hancock, P. J. (2012). Recovering faces from memory: the distracting influence of external facial features. *Journal of Experimental Psychology: Applied*, 18(2), 224. <https://doi.org/10.1037/a0027393>
- [10] Frowd, C. D., Pitchford, M., Skelton, F., Petkovic, A., Prosser, C., & Coates, B. (2012). Catching even more offenders with EvoFIT facial composites. *IEEE Third International Conference on Emerging Security Technologies (EST)*, 2012. pp. 20-26. <https://doi.org/10.1109/est.2012.26>
- [11] Ellis, H. D., Shepherd, J. W., & Davies, G. M. (1979). Identification of familiar and unfamiliar faces from internal and external features: Some implications for theories of face recognition. *Perception*, 8(4), 431-439. <https://doi.org/10.1068/p080431>
- [12] Frowd, C., Bruce, V., McIntyre, A., & Hancock, P. (2007). The relative importance of external and internal features of facial composites. *British Journal of Psychology*, 98(1), 61-77. <https://doi.org/10.1348/000712606x104481>
- [13] Frowd, C., Park, J., McIntyre, A., Bruce, V., Pitchford, M., Fields, S., Kenirons, M. & Hancock, P. J. (2008). Effecting an improvement to the fitness function. How to evolve a more identifiable face. *IEEE ECSIS Symposium on Bio-inspired Learning and Intelligent Systems for Security (BLISS'08)*, 2008. pp. 3-10. <https://doi.org/10.1109/bliss.2008.28>
- [14] Hancock, P. J., Bruce, V., & Burton, A. M. (2000). Recognition of unfamiliar faces. *Trends in Cognitive Sciences*, 4(9), 330-337. [https://doi.org/10.1016/s1364-6613\(00\)01519-9](https://doi.org/10.1016/s1364-6613(00)01519-9)
- [15] Tanaka, J. W., & Sengco, J. A. (1997). Features and their configuration in face recognition. *Memory & Cognition*, 25(5), 583-592. <https://doi.org/10.3758/bf03211301>
- [16] Frowd, C. D., Bruce, V., Plenderleith, Y., & Hancock, P. J. B. (2006). Improving target identification using pairs of composite faces constructed by the same person. *IET Conference on Crime and Security*, 2006. pp. 390-395. <https://doi.org/10.1049/ic:20060341>

- [17] Little, A. C., & Hancock, P. J. (2002). The role of masculinity and distinctiveness in judgments of human male facial attractiveness. *British Journal of Psychology*, 93(4), 451-464. <https://doi.org/10.1348/000712602761381349>
- [18] Kemp, R., Pike, G., White, P., & Musselman, A. (1996). Perception and recognition of normal and negative faces: The role of shape from shading and pigmentation cues. *Perception*, 25(1), 37-52. <https://doi.org/10.1068/p250037>
- [19] Yip, A. W., & Sinha, P. (2002). Contribution of color to face recognition. *Perception*, 31(8), 995-1003. <https://doi.org/10.1068/p3376>
- [20] Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1), 71-86.
- [21] Craw, I., & Cameron, P. (1991). Parameterising images for recognition and reconstruction. *British Machine Vision Conference*. Springer London. pp. 367-370. <https://doi.org/10.5244/c.5.52>
- [22] Hancock, P. J., Burton, A. M., & Bruce, V. (1996). Face processing: Human perception and principal components analysis. *Memory & Cognition*, 24(1), 26-40. <https://doi.org/10.3758/bf03197270>
- [23] Bruce, V., Hanna, E., Dench, N., Healey, P., & Burton, M. (1992). The importance of 'mass' in line drawings of faces. *Applied Cognitive Psychology*, 6(7), 619-628. <https://doi.org/10.1002/acp.2350060705>
- [24] O'Toole, A. J., Vetter, T., Blanz, V. (1999) Three-dimensional shape and two-dimensional surface reflectance contributions to face recognition: An application of three-dimensional morphing. *Vision Research*, 39, 3145-3155. [https://doi.org/10.1016/s0042-6989\(99\)00034-6](https://doi.org/10.1016/s0042-6989(99)00034-6)
- [25] Sinha, P., Balas, B. J., Ostrovsky, Y. & Russell, R. (2006). Face recognition by humans. In Zhao, W. and Chellappa, R. (Eds.), *Face processing: Advanced modeling and methods*, Amsterdam: Elsevier/Academic Press, 257-292
- [26] Shah, M. (1997). Fundamentals of computer vision (Unpublished manuscript). University of Central Florida.
- [27] Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6, 679-698. <https://doi.org/10.1109/tpami.1986.4767851>
- [28] Pérez, P., Gangnet, M., & Blake, A. (2003). Poisson image editing. *ACM Transactions on Graphics*, 22(3), 313-318. <https://doi.org/10.1145/882262.882269>
- [29] Garcia-Zurdo, R. (2016). Evolutive gradient face compositing using the Poisson equation. *Perception*, 45(2), 25-26.
- [30] Burton, A. M., White, D., & McNeill, A. (2010). The Glasgow face matching test. *Behavior Research Methods*, 42(1), 286-291. <https://doi.org/10.3758/brm.42.1.286>
- [31] Kazemi, V., & Sullivan, J. (2014). One millisecond face alignment with an ensemble of regression trees. *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. pp. 1867-1874. <https://doi.org/10.1109/cvpr.2014.241>
- [32] Liu, J., Mei, K., Ge, C., & Zheng, N. (2011). Interactive Poisson photometric propagation for facial composite. *1st International Symposium on Access Spaces (ISAS)*, 2011, pp. 121-126. <https://doi.org/10.1109/isas.2011.5960932>
- [33] Riviere, M., Teytaud, O., Rapin, J., LeCun, Y. and Couprie, C. (2019). Inspirational adversarial image generation. arXiv:1906.11661.

## Appendix. Gradient integration by solving Poisson's equation

The integration of the gradient of an image in order to get its corresponding intensity values reduces to the classic Poisson equation:

$$\Delta\varphi = f \quad (1)$$

where  $\Delta$  (read as "Del") stands for the Laplace operator or Laplacian. This expression means that the Laplacian of a certain unknown function equals  $f$ . The Laplacian is defined as the divergence of the gradient or as the sum of all unmixed second partial derivatives (the trace of the Hessian):

$$\Delta f = \nabla^2 f = \text{tr}(H) \quad (2)$$

Here,  $\nabla$  stands for the gradient operator:

$$\nabla f = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) \quad (3)$$

For a discrete 2D function  $I$ , the gradient may be approximated as a pair of forward finite differences in the  $x$  and  $y$  directions:

$$\begin{aligned} \nabla I &= (I_x, I_y) \\ I_x &= I(x+1, y) - I(x, y) \\ I_y &= I(x, y+1) - I(x, y) \end{aligned} \quad (4)$$

and the Laplacian can be calculated as the sum of the second-order unmixed gradients:

$$\begin{aligned} \Delta I &= I_{xx} + I_{yy} \\ I_{xx} &= I_{x+1} - I_x \\ I_{yy} &= I_{y+1} - I_y \end{aligned} \quad (5)$$

That is, the Laplacian of an image may be obtained from the sum of the horizontal gradient of the horizontal gradient plus the vertical gradient of the vertical gradient. By simple element arrangement, we arrive at the following finite difference scheme for the Laplacian:

$$\Delta I = I(x-1, y) + I(x+1, y) + I(x, y-1) + I(x, y+1) - 4I(x, y) \quad (6)$$

Now, we can set up a system of linear equations relating the known Laplacian of the image to the previous Laplacian scheme applied to the unknown pixel values. For each pixel in the image, an equation will be used of the form:

$$\begin{aligned} [\dots, 1, \dots, 1, -4, 1, \dots, 1, \dots] [x_1, x_2, \dots, x_n]^T \\ = [f_1, f_2, \dots, f_n]^T \end{aligned} \quad (7)$$

Here the left vector corresponds to a weight vector implementing the Laplacian scheme, the next vector on

the left side of the equation includes the unknown pixel values in the image, and the vector to the right of the equation includes the known Laplacian values calculated from the horizontal and vertical gradients. It should be noted that the 2D image has been flattened to a 1D vector.

The system of equations needs a boundary condition in order to obtain a solution (up to an additive factor), so we specify the values along the boundary of the domain (image area). This is known as a Dirichlet boundary condition. More specifically, a constant value is used for the boundary pixels. For each of these pixels, the weight values will all be zero, except for the one corresponding to the pixel position, which equals one.

$$[0, 0, \dots, 0, 1, 0, \dots, 0][x_1, x_2, \dots, x_n]^T = k \quad (8)$$

By stacking all the individual equations together, a linear system of equations is formed:

$$\mathbf{A}\mathbf{X} = \mathbf{B} \quad (8)$$

Here,  $\mathbf{A}$  is the weight matrix,  $\mathbf{X}$  is the unknown and  $\mathbf{B}$  is the known Laplacian and constant values matrix. This kind of system is sparse, because most of the elements in  $\mathbf{A}$  are zero, and therefore cannot be solved by ordinary means as the pseudo-inverse method. Instead, iterative solving methods such as Gauss-Seidel or Jacobi are used. In order to improve the solving speed, coarse-to-fine (also known as multigrid) methods may be used.

# Design Optimization Average-Based Algorithm

João Barradas Cardoso and Artur Barreiros

Department of Mechanical Engineering, Instituto Superior Técnico, University of Lisbon

Av Rovisco Pais 1, 1049-001 Lisbon, Portugal

E-mail: barreiros@tecnico.ulisboa.pt, <https://tecnico.ulisboa.pt/en/>

**Keywords:** heuristic, randomness, population-based, optimization

**Received:** October 6, 2017

*This article introduces a metaheuristic algorithm to solve engineering design optimization problems. The algorithm is based on the concept of diversity and independence that is aggregated in the average design of a population of designs containing information dispersed through a variety of points, and on the concept of intensification represented by the best design. The algorithm is population-based, where the population individual designs are randomly generated. The population can be normally or uniformly generated. The algorithm may start either with points randomly generated or with a designer preferred trial guess. The algorithm is validated using standard classical unconstrained and constrained engineering optimum design test problems reported in the literature. The results presented indicate that the proposed algorithm is a very simple alternative to solve this kind of problems. They compare well with the analytical solutions and/or the best results achieved so far. Two constrained problem analytical solutions not found in the literature are presented in annex.*

*Povzetek: Članek uvaja metahevristični algoritem za reševanje problemov optimizacije. Algoritem temelji na populaciji, kjer se naključno generirajo posamezni modeli posameznika in končno agregirajo v primerne rešitve.*

## 1 Introduction

With the advent of fast, cheap and reliable computing power over the last decades, in addition to the application of classic optimization to larger and larger size problems, new alternative algorithms operating in a different fashion have been developed. The classical optimization algorithms have shortcomings and are not suitable for all optimization problems. The new alternative algorithms allow attacking optimization problems either too costly or not applicable to classical algorithms.

The purpose of heuristic algorithms applied to optimization problems is to search a solution to them by trial-and-error in a satisfying amount of computing time. The optimum solution is not guaranteed but a near optimum solution is accepted as a good solution. Metaheuristics refers to higher-level algorithms combining lower-level techniques and tactics for exploration and exploitation of the design space. That is, these algorithms, on one hand, must be able to generate a range of points in the whole design space including potentially optimum ones; on the other hand, they intensify the search around the neighborhood of an optimum or near optimum points (Yang, 2008).

Exploration and exploitation are the two important components of the metaheuristic algorithms. They are also called diversification and intensification (Glover, Laguna, 1997). A good balance of these components is required. Too much weight in diversification risks slow convergence with the solutions jumping around the potentially optimum ones; too much weight in intensification restricts the design space to a local region and risks the convergence to a local optimum (Blum,

Roli, 2003). The heuristic algorithms start typically either with guess solutions randomly generated or with a designer preferred trial solution. The diversification is gradually reduced as the algorithm proceeds; simultaneously, the intensification is increased.

One of the roles of injected randomness in stochastic search is to allow for movements to unexplored areas of the search space that may contain an unexpectedly good design. This is especially relevant when the search is stalled near a local solution. Injected randomness may also be used for the creation of simple random quantities that act like their deterministic counterparts, but which are much easier to obtain and more efficient to compute.

Metaheuristics algorithms are either population-based or trajectory-based. Examples of the former ones are the genetic algorithms (Holland, 1975) or the particle swarm optimization (Cagnina, Esquivel, Coello, 2008; Kayham, Ceylan, Ayvaz, Gurarslan, 2010) the last ones include the simulated annealing optimization (Kirkpatrick, Gelatt, Vecchi, 1983) or the harmonic search (Geem, Kim, Loganathan, 2001).

If a sufficient numerous and diversified group of people is asked to decide on subjects of general interest, the decisions of the group are better than the decisions that an isolated individual would take (Surowiecki, 2005). Well informed and sophisticated an expert is, his or her advice and predictions should be pooled with those of others to get the most out of him or her. Practical examples, simple and complex, are described in (Surowiecki, 2005), remarking the principle of group think, and the concept that the masses are better problem

solvers, forecasters, and decision makers than any one individual. A classic example of group intelligence is the jelly-beans-in-the-jar experiment, in which invariably the group's average estimate for the number of the jelly beans in the jar is superior to the vast majority of the individual guesses.

The theory that groups are remarkably intelligent and often smarter than the smartest people in them, demonstrates the significant impact on how businesses operate, how knowledge is increased, how economies are structured, and how people live their daily lives (Williams, 2006). The necessary conditions for a crowd to be wise include diversity, independence, and a specific type of decentralization. These conditions are essential to making good decisions which are the result of disagreement and contest rather than consensus or compromise.

Diversity means individuals have some private information or their own interpretation of known facts. Diversity helps because it actually adds perspectives that would otherwise be absent and because it takes away, or at least weakens, some of the destructive characteristics of group decision-making. Homogeneous groups are great at doing what they do well, but they become progressively less able to investigate alternatives. Independence means freedom from the influence of others. It keeps the individual mistakes from becoming correlated. Diversity is essential to preserving this independence. In the jelly-beans-in-the-jar experiment, most group members are not talking to each other or solving problems together.

Decentralization means people draw on local knowledge. It encourages individuals to make important decisions, not just in one location based only on one specific type of information, but dispersed through a variety of locations from where local knowledge is drawn and shared. The information coming out of a decentralized group must be aggregated throughout the system, to maintain a balance between local and global counterparts. Aggregation needs a mechanism that turns individual judgments into a collective decision. For instance, in a free market, the aggregating mechanism is price; in the jelly-beans-in-the-jar experiment, individual guesses were aggregated and then averaged, i.e., the aggregating mechanism is the average guess.

The aim of the present article is the proposal of an algorithm based on the concept just described. The algorithm considered is stochastic in the sense that it relies on random numbers and that different results may be obtained upon running the algorithm repeatedly. The algorithm is population-based, where the population individual designs are randomly generated. These individuals are diversified and independent since their design variables values are chosen stochastically without any correlation.

The individual designs are also decentralized since the design variables are chosen all over the entire design space. Finally, the different values of the design variables are aggregated as the plain or weighted average of those values. However, generating a diverse set of possible solutions isn't enough. The designer, as the body of

people, also has to be able to distinguish the good solutions for the bad. So, at each of the iterations, the present algorithm selects two designs: the best design, the one with the best objective so far; and the averaged design, the one which design variable values are the mean of those variable values for the iteration.

The best design represents the intensification component of the algorithm; the averaged design represents the diversity part. In the current article, the optimization problem is understood as a minimization problem, where the function of merit to be assessed is an extended cost function that takes in account penalization due to violated constraints. A reference design is considered as a linear combination of both the best and the averaged design variables. A simple recurrence formula, centered on the reference design, is used to actualize the design for the next randomly generated population. The population can be normally or uniformly generated.

In optimization, there is traditionally a concern with developing a good stopping criterion. Unfortunately, the quest for an automatic means of stopping an algorithm with a guaranteed level of accuracy seems doomed to failure in general stochastic search problems. The fundamental reason is that in nontrivial problems, there will always be a significant region within the design space that will remain unexplored in any finite number of iterations, and there is always the possibility the optimum could lie in this unexplored region.

A danger arises in making broad claims about the performance of an algorithm based on the results of numerical studies. Performance can vary tremendously under even small changes in the form of the functions involved or the coefficient settings within the algorithms themselves. Outstanding performance on some types of functions is consistent with poor performance on other types of functions. This is a manifestation of no free lunch theorems (Spall, 2003; Wolpert, Macready, 1997).

The present algorithm is applied to several constrained and unconstrained test functions as well as to typical engineering design problems.

## 2 The optimal design problem

The optimal design problem may be formulated in a generalized fashion as

$$\begin{aligned} & \min_{\mathbf{b}} \Psi_0(\mathbf{b}) \\ & s.t. \\ & \Psi_j(\mathbf{b}) \leq 0; \quad j = 1, 2, \dots, m \\ & b_{il} \leq b_i \leq b_{iu}; \quad i = 1, 2, \dots, n \end{aligned} \quad (1)$$

where  $\Psi_0$  is the cost or objective function,  $\Psi_j$  are the constraints in a dimensionless form,  $\mathbf{b} \equiv (b_1, b_2, \dots, b_n)$  is the design vector,  $b_{il}$  and  $b_{iu}$  are respectively the lower and upper bounds of the design variable  $b_i$ .

In order of applying the present algorithm, the formulation of the Eq. (1) is replaced by the following one:

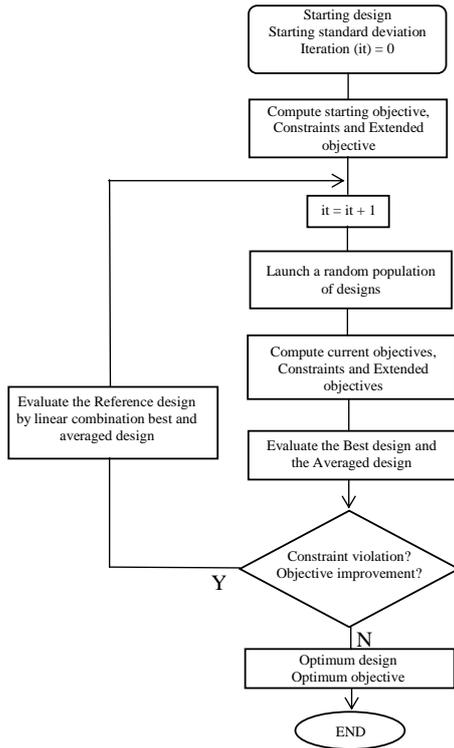


Figure 1: Flowchart of the algorithm.

$$\begin{aligned} \min_b \bar{\Psi}_0 &\equiv \Psi_0 + |\Psi_0| P \\ \text{s.t.} \quad & b_i \leq b_l \leq b_{iu}; \quad i = 1, 2, \dots, n \end{aligned} \quad (2)$$

In Eq. (2),  $P$  stands for the penalty factor which value depends on the violation of the constraints as

$$P = \begin{cases} 0, & \text{for satisfied constraints} \\ \sum_{v=1}^{m'} \alpha \Psi_v / m', & \text{for violated constraints} \end{cases} \quad (3)$$

where  $\alpha > 0$  and  $m'$  is the number of violated constraints. In practical terms, the constraint  $\Psi_j$  can be considered violated if  $\Psi_j > \varepsilon$ , with  $\varepsilon$  a very small number. Note that the absolute value of the objective is considered in the Eq. (2) in order to accommodate problems with negative objective functions.

$\theta$	Starting Point			
	(-2.5, -2.5)	(-2.5, 2.5)	(2.5, -2.5)	(2.5, 2.5)
1.00	(0.08986, -0.71283) -1.0316283 at iteration 38	(0.09140, -0.71266) -1.0316191 at iteration 39	(-0.09162, 0.71276) -1.0316163 at iteration 36	(0.08887, -0.71247) -1.0316248 at iteration 31
0.85	(0.08979, -0.71267) -1.0316285 at iteration 15	(0.08997, -0.71265) -1.0316285 at iteration 21	(-0.08991, 0.71265) -1.0316285 at iteration 21	(-0.08976, 0.71272) -1.0316285 at iteration 15
0.25	(0.08973, -0.71262) -1.0316285 at iteration 36	(-0.08992, -0.71259) -1.0316285 at iteration 29	(0.08974, -0.71262) -1.0316285 at iteration 26	(-0.08973, 0.71262) -1.0316285 at iteration 37
0.00	(0.09002, -0.71401) -1.0316136 at iteration 989	(-0.09020, 0.71196) -1.0316237 at iteration 617	(0.09020, -0.71196) -1.0316237 at iteration 617	(-0.09002, 0.71401) -1.0316136 at iteration 989

 Table 1: Optimum points and function values for different  $\theta$ s and starting points.

### 3 The average concept algorithm

The present algorithm may be described by the flowchart of Fig. 1 as well as in the following manner:

1. Start with a preference design guess  $\mathbf{b}$  and an imposed standard deviation, estimated as  $s_i = b_{iu} - b_{il}$  ( $i = 1, 2, \dots, n$ ). Compute the starting values of  $\Psi_0$ ,  $\Psi_j$  ( $j = 1, 2, \dots, m$ ) and  $\bar{\Psi}_0$ , and establish the starting reference design  $\mathbf{b}^R$  as the preference guess, i.e.  $\mathbf{b}^R \equiv \mathbf{b}$ .

Start iterates as

2. Launch a normally or uniformly random population of  $N$  designs as

$$b_i^k = b_i^R + s_i r_i^k \quad (4)$$

where  $i = 1, 2, \dots, n$ ,  $k = 1, 2, \dots, N$  and  $r_i^k$  is the  $k$ -th random number (with mean value 0 and standard deviation 1) related to the design variable  $b_i$ .

Do  $b_i = b_{il}$  if  $b_i < b_{il}$  and  $b_i = b_{iu}$  if  $b_i > b_{iu}$ .

3. Evaluate  $\Psi_0$ ,  $\Psi_j$  ( $j = 1, 2, \dots, m$ ) and  $\bar{\Psi}_0$  for the entire population of  $N$  designs.
4. Evaluate the best design  $\mathbf{b}^B$  corresponding to the minimum value  $\bar{\Psi}_0^{\min}$  of the extended function.

Evaluate the averaged design  $\mathbf{b}^A$  for the distribution of designs as

$$\mathbf{b}^A = \sum_{k=1}^N \mathbf{b}^k p_k / \sum_{k=1}^N p_k \quad (5)$$

where  $\mathbf{b}^k$  is an arbitrary design vector in the population and  $p_k$  is the weight accounted for design  $\mathbf{b}^k$  on the average. The weights are selected by the designer. If a plain average is chosen, then  $p_k = 1$  for every design.

Another choice used in this work is to compare the

values of the extended function  $\bar{\Psi}_0$  and of the best extended function  $\bar{\Psi}_0^{\min}$  in the previous iteration; and then assign a weight  $p_k = 2$  if the first value is smaller than the second one, and a weight  $p_k = 1$  if it is equal or larger.

5. If there are no constraint violations and there are no improvements of the objective function within a prescribed number of iterations, go to 7.
6. Evaluate a reference design as the linear combination

$$\mathbf{b}^R = \theta \mathbf{b}^B + (1 - \theta) \mathbf{b}^A \tag{6}$$

with  $0 \leq \theta \leq 1$  and assume the distribution of designs  $\mathbf{b}^k$  centered at  $\mathbf{b}^R$  with a standard deviation vector evaluated as  $\mathbf{s} = 2|\mathbf{b}^A - \mathbf{b}^B|$ .

Go to 2.

End the iterates.

7. Stop.

In order of handling tabular discrete value design variables, the Eq. (4) is rewritten in these cases as

$$b_i^k = \text{int} \left( \frac{b_i^R + s_i r_i^k}{\Delta} \right) \Delta \tag{7}$$

where  $\Delta$  is the difference value between two consecutive design variables.

## 4 Numerical applications

In this Section one is going to solve unconstrained as well as constrained optimization problems by applying the formulation and the algorithm presented on the previous sections. With respect to the unconstrained problems, the applications are the minimizations of the benchmark following test functions: Six-Hump Camelback function, Rosenbrock and Michalewicz’s functions.

Concerning to the constrained problems, one solves three well-known engineering design optimization test problems: the welded beam design, the pressure vessel design and the tension-compression spring design. For all the applications normally distributed populations were used. A total of 30 independent runs were performed per problem. In the Sections 4.1 to 4.6 the runs of the algorithm were performed with a particular initial seed for different population sizes and different weights on the calculation of the average and reference designs.

The results are compared with analytical solutions and/or heuristic and nonlinear programming algorithms.

### 4.1 Six-Hump Camelback function

This function is one of the typical test functions in unconstrained global optimization (Dixon, Szego, 1975; Lee, Geem, 2005). It is mathematically expressed as

$$\Psi_0 = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4 \tag{8}$$

Within the bounded region, this function has six local minima. Two of them are global minima located at either  $(-0.08984, 0.71266)$  or  $(0.08984, -0.71266)$ , each with the corresponding function value equal to  $\Psi_0^* = -1.0316285$ .

For the algorithm presented here, a population of size  $N = 20$  and a plain average in the step 4 were used within the design space  $-2.5 \leq x_1, x_2 \leq 2.5$ . For different values of the  $\theta$ -factor and two starting points, the Table 1 gives the corresponding achieved solutions and the number of iterations needed for convergence.

We should note that the best and fastest solutions are obtained with  $\theta = 0.85$ , i.e., giving respectively at each iteration, the weights 0.85 and 0.15 to the best and average points of the prior iteration in the formation of the current reference design.

With the starting point  $(2.5, 2.5)$ , corresponding to the cost value  $\Psi_0 = 161.8489685$ , and with  $\theta = 0.85$ , the present algorithm achieves the minimum cost value  $\Psi_0^* = -1.0316285$  at the point  $(-0.08976, 0.71272)$  after 15 iterations. However, after 9 iterations, the algorithm gets the cost value  $\Psi_0 = -1.0316285$  at the point  $(-0.08799, 0.71382)$ , not far off the analytical solution.

We should also note that the worst results exist for  $\theta = 0$ , i.e., only considering the average design as the design of reference. For  $\theta = 1$ , respecting to the selection at each iteration of the best design as the reference one, the solutions are also not so good

For  $N = 20$ ,  $\theta = 0.85$ , starting point  $(2.5, 2.5)$  and the weighted average (second) option on step 4 described in Chapter 3 instead of the plain average, the global optimum  $\Psi_0^* = -1.0316285$  is obtained at the point  $(-0.08988, 0.71266)$  after 19 iterations.

Better results can be expectedly obtained by increasing the population size. For  $N = 100$ , the best and fastest solutions are obtained with  $\theta = 0.85$  after 10 iterations at the points  $(-0.08982, 0.71264)$  and  $(0.08983, -0.71264)$  respectively starting from the points  $(-2.5, 2.5)$  and  $(2.5, -2.5)$ , corresponding both to the minimum function value  $\Psi_0^* = -1.0316285$ . For

$N$	$IP$	Iterate	$x_1$	$x_2$	$\Psi_0$
1000	0	4	-0.0897507	0.7126449	-1.0316285
	1	5	-0.0898749	0.7127146	-1.0316285
100	0	13	-0.0897433	0.7126887	-1.0316285
	1	11	-0.0898545	0.7126102	-1.0316285
20	0	15	-0.0897623	0.7127175	-1.0316285
	1	19	-0.0898786	0.7126648	-1.0316285

Table 2: Camelback’s optimal solutions for  $\theta = 0.85$  and starting at  $(2.5, 2.5)$ .

$N=1000$  and  $\theta=0.85$ , and starting from the point  $(-2.5, -2.5)$ , the optimum solution  $\mathbf{x}^* = (0.08975, -0.71264)$ ,  $\Psi_0^* = -1.0316285$  has been achieved after 4 iterations.

Table 2 shows the optimal solutions achieved for different N population sizes and different average choices on Eq. (5),  $IP=0$  standing for plain average and  $IP=1$  meaning weighted average as written on step 4 of the algorithm described on Chapter 3, using  $\theta=0.85$  and with the starting point located at  $(2.5, 2.5)$ .

We may note that is the population size  $N=20$  the one that needs the smallest number of function evaluations  $(20 \times 15)$ .

### 4.2 Rosenbrock’s function

Another classical test function in unconstrained optimization is the Rosenbrock’s function, which two-dimensional form (Moré, Garbow, Hillstrom, 1981; Rosenbrock, 1960; Yang, 2008) is

$$\Psi_0 = (1 - x_1)^2 + 100(x_2 - x_1^2)^2 \quad (9)$$

This function also referred to as the Valley or Banana function due to the shape of its contour lines, is a popular test problem for gradient-based optimization algorithms. The function turns out to be quite challenging to find its minimum point by numerical methods.

Its global optimum point is  $\mathbf{x}^* = (1.0, 1.0)$  that gives the optimum cost of  $\Psi_0^* = 0.0$ . The function is unimodal and its analytical solution can be obtained straightforwardly by partial differentiation. The numerical solution, however, poses a particular challenge. The solution lies inside a very deep, narrow, banana shaped valley. The valley causes a lot of troubling for nonlinear programming search algorithms.

Using a population of 1000 samples within the design space  $-10.0 \leq x_1, x_2 \leq 10.0$ , a plain average, the factor  $\theta=0.85$  and a starting point  $\mathbf{x} = (0.0, 0.0)$  corresponding to  $\Psi_0 = 1.0000000$ , the present algorithm achieves the optimum point  $\mathbf{x}^* = (1.00000, 1.00000)$  and the corresponding  $\Psi_0^* = 0.0000000$  after 18 iterations. However, the point  $\mathbf{x} = (1.00051, 1.00087)$  corresponding to  $\Psi_0^* = 0.0000029$  was obtained after 9 iterations. If  $N=100$ , the same optimum point above is obtained after 38 iterations. For  $N=20$ , the algorithm converges towards the same optimum point after 2604 iterations. One may say that is the population of 100 samples that gives the shortest number of function evaluations  $(100 \times 38)$ .

### 4.3 Michalewicz’s function

The third unconstrained optimization problem uses the Michalewicz’s function in its two-dimensional form

$$\Psi_0 = -\sum_{i=1}^2 \sin(x_i) \left[ \sin(i x_i^2 / \pi) \right]^{20} \quad (10)$$

with  $0 \leq x_1, x_2 \leq \pi$ .

The function is tricky; it has several local minimum values and several flat areas which make the one global minimum value hard to find numerically. Its global minimum is  $\Psi_0^* = -1.8012980$  at the point  $\mathbf{x}^* = (2.20319, 1.57049)$ .

Using a population with size  $N=20$  samples within the design space  $0.0 \leq x_1, x_2 \leq 4.0$ , a plain average, the factor  $\theta=0.85$ , and a starting point  $\mathbf{x} = (2.0, 2.0)$  corresponding to  $\Psi_0 = -0.3701513$ , the present algorithm achieves the optimum point  $\mathbf{x}^* = (2.20281, 1.57079)$  and the corresponding cost value  $\Psi_0^* = -1.8013037$  after 24 iterations. However, after 13 iterations the algorithm achieves the point  $\mathbf{x} = (2.20258, 1.56925)$  corresponding to the objective value  $\Psi_0 = -1.8012055$ . Changing the population size to  $N=100$ , the global minimum  $\Psi_0^* = -1.8013039$  is achieved after 10 iterations at the point  $\mathbf{x}^* = (2.20289, 1.57091)$ . Changing now the population size to  $N=1000$ , the global minimum  $\Psi_0^* = -1.8013039$  is achieved after 8 iterations at the point  $\mathbf{x}^* = (2.20292, 1.57076)$ . The smallest number of function evaluations  $(20 \times 24)$  is obtained for  $N=20$ .

### 4.4 Welded beam design

The welded beam design problem is well studied in the context of single-objective optimization. A beam A needs to be welded on another beam B and must carry a certain load P as shown in Fig. 2. The welded beam is designed for minimum fabrication cost subject to constraints on

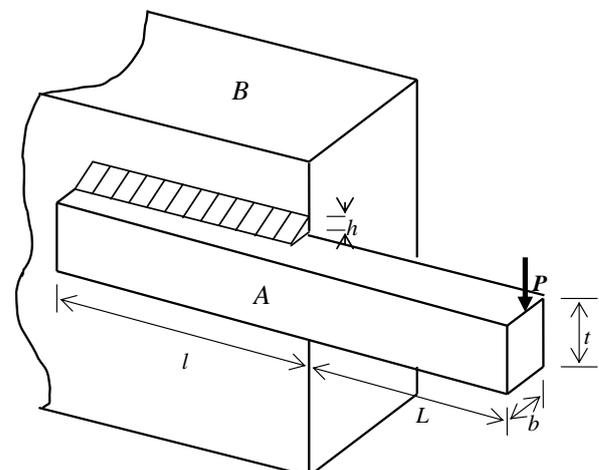


Figure 2: Welded beam structure.

shear stress  $\tau$ , bending stress in the beam  $\sigma$ , buckling load on the bar  $P_c$ , end deflection of the beam  $\delta$ , cost of the weld and beam  $A$  materials, and side constraints (Ragsdell, Phillips, 1975; Rao, 1996). One wants to find four design parameters: thickness of the beam  $b$ , width of the beam  $t$ , length of the weld  $l$ , and thickness of the weld  $h$ .

In order of formulating the problem in a standard form, let  $\mathbf{x} \equiv (x_1, x_2, x_3, x_4) = (h, l, t, b)$  be the design vector. Then, our problem may be described as

$$\begin{aligned} \min \Psi_0 &\equiv (C_1 + C_2)x_1^2 x_2 + C_3 x_3 x_4 (L + x_2) \\ \text{s.t.} \\ \Psi_1 &\equiv \tau / \tau_{\max} - 1 \leq 0 \\ \Psi_2 &\equiv \sigma / \sigma_{\max} - 1 \leq 0 \\ \Psi_3 &\equiv x_1 / x_4 - 1 \leq 0 \\ \Psi_4 &\equiv [C_1 x_1^2 + C_3 x_3 x_4 (L + x_2)] / 5 - 1 \leq 0 \quad (11) \\ \Psi_5 &\equiv \delta / \delta_{\max} - 1 \leq 0 \\ \Psi_6 &\equiv 1 - P_c / P \leq 0 \\ 0.125 &\leq x_1 \leq 2 \\ 0.1 &\leq x_4 \leq 2 \\ 0.1 &\leq x_1, x_2 \leq 10 \end{aligned}$$

where  $C_1 = 0.10471 \text{ \$/in}^3$  is the cost per unit volume of

the weld material,  $C_2 = 1 \text{ \$/in}^3$  is the labor cost per unit weld volume,  $C_3 = 0.04811 \text{ \$/in}^3$  is the cost per unit volume of the beam  $B$ ,  $\tau_{\max} = 13,600 \text{ psi}$ ,  $\sigma_{\max} = 30,000 \text{ psi}$  and  $\delta_{\max} = 0.25 \text{ in}$ . The other parameters are defined as:

$$\begin{aligned} \tau &= \sqrt{\tau_1^2 + \tau_1 \tau_2 x_2 / R + \tau_2^2}, \quad \tau_1 = P / (\sqrt{2} x_1 x_2), \\ \tau_2 &= M R / J, \quad M = P (L + x_2 / 2), \\ R &= 0.5 \sqrt{x_2^2 + (x_1 + x_3)^2}, \quad \sigma = 6 P L / (x_3^3 x_4), \\ \delta &= 4 P L^3 / (E x_3^3 x_4), \quad J = (\sqrt{2} / 6) x_1 x_2 [x_2^2 + 3(x_1 + x_3)^2], \\ P_c &= 4.013 (E / L^2) (x_3 x_4^3 / 6) [1 - (0.5 x_3 / L) \sqrt{0.25 E / G}], \\ E &= 30 \times 10^6 \text{ psi}, \quad G = 12 \times 10^6 \text{ psi}, \quad P = 6,000 \text{ lb}, \\ L &= 14 \text{ in} \end{aligned}$$

By using mathematical programming, (Rao, 1996) presents the optimum cost function  $\Psi_0^* = 2.3810$  corresponding to the design point  $\mathbf{x}^* = (0.2444, 6.2177, 8.2915, 0.2444)$ . A better nonlinear programming solution has been achieved in (Andrei, 2013) by adding GAMS created nonlinear model:  $\Psi_0^* = 1.72485$ ,  $\mathbf{x}^* = (0.206, 3.470, 9.037, 0.206)$ .

The lowest optimal solution known so far by the

N	IP	Iterate	$x_1$	$x_2$	$x_3$	$x_4$	$\Psi_0$
5000	0	16645	0.205729	3.470519	9.036630	0.205730	1.724858
		268	0.205468	3.476687	9.038588	0.205723	1.725573
		32	0.205436	3.477702	9.034963	0.205964	1.726863
		16	0.204191	3.530001	9.042473	0.205769	1.731815
	1	11144	0.205726	3.470586	9.036639	0.205730	1.724864
		303	0.204957	3.488440	9.035478	0.205797	1.726386
		57	0.205409	3.475931	9.042488	0.205808	1.726697
		19	0.204431	3.523721	9.041501	0.205707	1.730704
1000	0	9434	0.205727	3.470567	9.036656	0.205730	1.724866
		27	0.204332	3.495317	9.051999	0.205779	1.729056
		19	0.204359	3.503654	9.084193	0.205596	1.734414
	1	10419	0.205730	3.470530	9.036634	0.205730	1.724866
		1335	0.205707	3.470647	9.038382	0.205760	1.725373
		522	0.205451	3.477778	9.038218	0.205742	1.725777
		37	0.205635	3.479632	9.039730	0.205804	1.727052
		21	0.206227	3.492924	9.013144	0.206816	1.732874
100	0	15080	0.205727	3.470617	9.036695	0.205729	1.724877
		1596	0.205693	3.470650	9.039433	0.205731	1.725315
		205	0.205585	3.477111	9.031481	0.206261	1.728674
		177	0.206150	3.503281	9.041299	0.206170	1.734152
	1	27689	0.205724	3.470646	9.036589	0.205731	1.724869
		672	0.204555	3.495600	9.037673	0.205735	1.726630
		300	0.205584	3.488843	9.039643	0.205969	1.729467
		141	0.202780	3.547630	9.027128	0.206246	1.732925
20	0	11462	0.205731	3.470480	9.036694	0.205735	1.724909
		2265	0.205546	3.475157	9.036914	0.205765	1.725515
		714	0.207129	3.459350	8.988936	0.208278	1.736547
	1	22011	0.205726	3.470606	9.036623	0.205730	1.724868
		2788	0.205607	3.474374	9.036784	0.205731	1.725223
		115	0.202686	3.587231	9.037897	0.205726	1.736020

Table 3: Welded beam optimization solutions.

$N$	$IP$	Iterate	$x_1$	$x_2$	$x_3$	$x_4$	$\Psi_0$
1000	0	4143	0.81250	0.43750	42.09843	176.63712	6059.7231
	1	13441	0.81250	0.43750	42.09843	176.63678	6059.7158
500	0	8285	0.81250	0.43750	42.09843	176.63701	6059.7212
	1	11909	0.81250	0.43750	42.09844	176.63673	6059.7168
200	0	4173	0.81250	0.43750	42.09838	176.63742	6059.7227
	1	5077	0.81250	0.43750	42.09843	176.63704	6059.7222

Table 4: Pressure vessel optimal solutions.

authors is given in (Kayham, Ceylan, Ayvaz, Gurarslan, 2010):  $\mathbf{x}^* = (0.205830, 3.468338, 9.036624, 0.205730)$  and  $\Psi_0^* = 1.724717$ .

The optimization results achieved with the present algorithm are shown in Table 3 for different population sizes  $N$  and different average choices on Eq. (5). The first row for each choice combination of  $N$  and  $IP$  represents the optimum at the convergence of the algorithm. The following rows are the results at intermediary iterations. For all these design points all the constraints are satisfied.

The best minimum cost value obtained in the present article is  $\Psi_0^* = 1.724858$ , corresponding to the point  $\mathbf{x}^* = (0.205729, 3.470519, 9.036630, 0.205730)$ .

One may observe that the algorithm solutions compare very well with the best solution presented above, even for earlier iterates of the algorithm. We may also observe that convergence is faster when the plain average is used in Eq. (5).

### 4.5 Pressure vessel design

The pressure vessel design problem has been proposed in (Kannan, Kramer, 1994). It is one of the most used test problems for validating optimization algorithms. The problem is to find the optimal design of a compressed air storage tank (Fig. 3) with a working pressure of 1000 psi and a minimum capacity volume of  $V_{\min} = 1,296,000 \text{ in}^3$ .

The pressure vessel is composed of a cylindrical shell capped at both ends by hemispherical heads.

Let the design variables be  $x_1 \equiv T_s$  the thickness of the shell,  $x_2 \equiv T_h$  the thickness of the heads,  $x_3 \equiv R$  the inner radius and  $x_4 \equiv L$  the length of the cylindrical shell. The variables  $x_1$  and  $x_2$  should be integer multiples of 0.0625 in. The objective is to minimize the manufacturing cost (material, welding and forming costs) of the pressure vessel (Sandgren, 1990), subjected to constraints on volume capacity and in accordance with

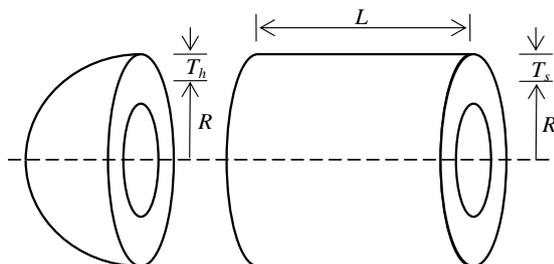


Figure 3: Pressure vessel.

respective ASME codes. The mathematical model of the problem is:

$$\min \Psi_0 \equiv 0.6224 x_1 x_3 x_4 + 1.7781 x_2 x_3^2 + 3.1661 x_1^2 x_4 + 19.84 x_1^2 x_3$$

s.t.

$$\Psi_1 \equiv 0.0193 x_3 / x_1 - 1 \leq 0$$

$$\Psi_2 \equiv 0.00954 x_3 / x_2 - 1 \leq 0 \tag{12}$$

$$\Psi_3 \equiv V_{\min} / V - 1 \leq 0$$

$$V = \pi (x_3^2 x_4 + 4 x_3^3 / 3)$$

$$0.0625 \leq x_1, x_2 \leq 99 \times 0.0625$$

$$10 \leq x_3, x_4 \leq 200$$

The analytical optimum for this problem is calculated in the Annex A as  $\Psi_0^* = 6059.714335$  at  $\mathbf{x}^* = (0.8125, 0.4375, 42.0984456, 176.6365958)$  with the first and third constraints being active.

The optimal results achieved with the present algorithm are shown in Table 4 for different population sizes  $N$  and different average choices on Eq. (5). For all these solutions there is no violation of the constraints. The first constraint is nearly active at the optimal point for all the different population sizes; it takes values in the interval  $-0.3666 \times 10^{-7} \leq \Psi_1^* \leq -0.1215 \times 10^{-5}$ .

The results for the present algorithm compare well with the analytical solution. Again, the plain average of the Eq. (5) gives origin to faster convergence.

### 4.6 Tension/Compression spring design

The tension/compression spring design optimization problem is described in (Arora, 1989). The goal is to minimize the weight of a tension/compression spring (Fig. 4) subject to constraints on minimum deflection, shear stress, surge frequency, limits on outside diameter and side constraints. The design variables to be considered are the wire diameter  $d$ , the mean coil diameter  $D$  and the number  $n$  of active coils.

Let us set up the vector of design variables as  $\mathbf{x} \equiv (x_1, x_2, x_3) \equiv (d, D, n)$ .

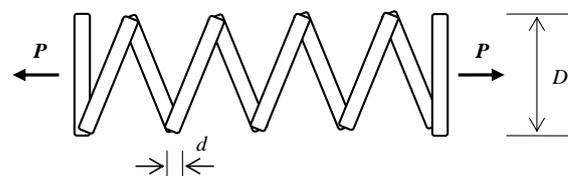


Figure 4: Tension-compression spring.

The problem may be formulated as

$$\begin{aligned}
 \min \Psi_0 &\equiv (x_3 + 2)x_1^2 x_2 \\
 \text{s.t.} \\
 \Psi_1 &\equiv 1 - x_2^3 x_3 / (71785 x_1^4) \leq 0 \\
 \Psi_2 &\equiv (4x_2^2 - x_1 x_2) / [12566(x_2 x_1^3 - x_1^4)] + \\
 &\quad 1 / (5108 x_1^2) - 1 \leq 0 \\
 \Psi_3 &\equiv 1 - 140.45 x_1 / (x_2^2 x_3) \leq 0 \\
 \Psi_4 &\equiv (x_1 + x_2) / 1.5 - 1 \leq 0 \\
 0.05 &\leq x_1 \leq 2 \\
 0.25 &\leq x_2 \leq 1.3 \\
 2 &\leq x_3 \leq 15
 \end{aligned} \tag{13}$$

The analytical solution for this problem is presented in the Annex B. The minimum weight of the spring is achieved as  $\Psi_0^* = 0.012665232$  at the point  $x^* = (0.051690, 0.356740, 11.287642)$ . At the optimum, the constraints  $\Psi_1$  and  $\Psi_2$  are active, and  $\Psi_3^* = -4.054$ ,  $\Psi_4^* = -0.7277$ .

The minimum objective function value obtained in (Arora, 1989) by nonlinear programming is  $\Psi_0^* = 0.012679$  corresponding to the optimum point  $x^* = (0.051699, 0.35695, 11.289)$ . The best result known so far by the authors is given in (Cagnina, Esquivel, Coello, 2008) as  $\Psi_0^* = 0.012665$ , with  $x^* = (0.051583, 0.354190, 11.438675)$ .

The optimum results achieved with the present algorithm are shown in Table 5, for  $IP = 0$ , different sizes  $N$  of the population and plain average selection on Eq. (5). For all these solutions there is no violation of the constraints, being practically active the first two constraints  $\Psi_1$  and  $\Psi_2$ . The last two constraints have values within the intervals  $-4.08 \leq \Psi_3^* \leq -4.04$  and  $-0.734 \leq \Psi_4^* \leq -0.720$ .

Again, the present algorithm results compare well with the analytical solution.

### 5 Concluding remarks

This article presents an average concept algorithm to solve various optimization problems which include typical benchmark functions unconstrained problems and structural engineering test design constrained problems.

$N$	1000	500	200	100
$\Psi_0$	0.012665	0.012666	0.012668	0.012671
$x_1$	0.051718	0.051775	0.051308	0.052127
$x_2$	0.357409	0.358778	0.347629	0.367321
$x_3$	11.248693	11.169669	11.842695	10.694988
Iterate	10063	11690	24818	13899

Table 5: Tension-compression spring optimal solutions.

To evaluate the performance of the present algorithm, numerical applications are conducted and the results are compared to the results obtained analytically and/or to the best ones achieved by other optimization methods. The analytical solutions for two of the constrained problems, namely the pressure vessel design and the tension-compression spring design problems, are determined in the present article. The solutions found by the proposed algorithm compare well with those results.

We may conclude that the algorithm finds the global solution or a near-global solution in each problem tested. The Six-Hump Camelback function, for example, has 6 local minimal points; however, the algorithm converges to the global minima.

Characterizing the principal advantage of the algorithm one should emphasize the good balance between the accuracy of the solutions it achieves and its rare simplicity.

### References

- [1] Andrei, N. (2013) Nonlinear Optimization Applications Using the GAMS Technology. Springer Science & Business Media. <https://doi.org/10.1007/978-1-4614-6797-7>
- [2] Arora, J. S. (1989) Introduction to Optimum Design. McGraw-Hill, New York. <https://doi.org/10.1016/C2013-0-15344-5>
- [3] Blum, C., Roli, A. (2003) Metaheuristics in Combinatorial Optimization: overview and conceptual comparison. ACM Computing Surveys. 35, 268-308. <https://dl.acm.org/doi/pdf/10.1145/937503.937505>
- [4] Cagnina, L.C., Esquivel, S.C., Coello, C.A.C. (2008) Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer, Informatica. 32, 319-326.
- [5] Dixon, L.C.W., Szego, G.P. (1975) Towards Global Optimization. North Holland.
- [6] Geem, Z.W., Kim, G.H., Loganathan, G.V. (2001) A New Heuristic Optimization Algorithm: Harmonic Search. Simulation. 76, 60-68. <https://doi.org/10.1177/003754970107600201>
- [7] Glover, F., Laguna, M. (1997) Tabu Search. Kluwer Academic Publishers. [https://doi.org/10.1007/978-1-4613-0303-9\\_33](https://doi.org/10.1007/978-1-4613-0303-9_33)
- [8] Holland, J.H. (1975) Adaptation in Natural and Artificial Systems. The University of Michigan Press: Ann Arbor.
- [9] Kannan, B.K., Kramer, S.N. (1994) An Augmented Lagrange Multiplier Based Method for Mixed Integer Discrete Continuous Optimization and its Applications to Mechanical Design. Journal of Mechanical Design. 116, 318-320. <https://doi.org/10.1115/1.2919393>
- [10] Kayham, A.H., Ceylan, H., Ayvaz, M.T., Gurarlan, G. (2010) PSOLVER: A New Hybrid Particle Swarm Optimization Algorithm for Solving Continuous Optimization Problems. Expert Systems with Applications. 37, 6798-6808. <https://doi.org/10.1016/j.eswa.2010.03.046>

[11] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. (1983) Optimization by Simulated Annealing. *Science*. 220, 671-680.  
<https://doi.org/10.1126/science.220.4598.671>

[12] Lee, K.S., Geem, Z.W. (2005) A New Meta-Heuristic Algorithm for Continuous Engineering Optimization: Harmony Search Theory and Practice. *Computer Methods in Applied Mechanical Engineering*. 194, 3902–3933.  
<https://doi.org/10.1016/j.cma.2004.09.007>

[13] Luenberger, D.G. (1984) *Introduction to Linear and Nonlinear Programming*. Addison-Wesley, Massachusetts.  
<https://doi.org/10.1007/978-0-387-74503-9>

[14] Moré, J.J., Garbow, B.S., Hillstom, K.E. (1981) Testing Unconstrained Optimization Software. *ACM Transactions on Mathematical Software*. 7, (1), 17- 41.  
<https://doi.org/10.1145/355934.355936>

[15] Papalambros, P.Y., Wilde, D.J. (1988) *Principles of Optimal Design: modeling and computation*. Cambridge University Press, New York.  
<https://doi.org/10.1017/9781316451038>

[16] Ragsdell, K. M., Phillips, D.T. (1975) Optimal Design of a Class of Welded Structures Using Geometric Programming. *ASME J. Eng. Ind. Ser. B*. 98 (3), 1021–1025.  
<https://doi.org/10.1115/1.3438995>

[17] Rao, S.S. (1996) *Engineering Optimization Theory: and Practice*. John Wiley & Sons.  
<https://doi.org/10.1002/9780470549124>

[18] Rosenbrock, H.H. (1960) An Automatic Method for Finding the Greatest or Least Value of a Function. *Computer Journal*. 3, 175-184.  
<https://doi.org/10.1093/comjnl/3.3.175>

[19] Sandgren, E. (1990) Nonlinear Integer and Discrete Programming in Mechanical Design Optimization. *Journal of Mechanical Design*. 112 (2), 223–229.  
<https://doi.org/10.1115/1.2912596>

[20] Spall, J.C. (2003) *Introduction to Stochastic Search and Optimization: estimation, simulation and control*. John Wiley&Sons.  
<https://doi.org/10.1002/0471722138>

[21] Surowiecki, J. (2005) *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*. Anchor Books.

[22] Williams, S. (2006) *The Wisdom of Crowds: why the many are smarter than the few and how collective wisdom shapes business, economies, societies, and nations*. *Business Book Review™*. 21 (43).

[23] Wolpert, D.H., Macready, W.G. (1997) No Free Lunch Theorems for Optimization. *IEEE Transactions on Evolutionary Computation*. 1, 67-82.  
<https://doi.org/10.1109/4235.585893>

[24] Yang, X.S. (2008) *Nature-inspired Metaheuristic Algorithms*. Luniver Press.

## Annex A: Pressure Vessel Classical Design Optimization

Let us assume initially the variables  $x_1$  and  $x_2$  are continuous and later on make the convenient correction to table (multiple of 0.0675) values.

The cost function  $\Psi_0$  decreases monotonically with  $x_1$  or  $x_2$  decrease. The constraint  $\Psi_1$  is the only constraint that increases as  $x_1$  decreases, and the constraint  $\Psi_2$  is the only constraint that increases as  $x_2$  decreases. Then,  $\Psi_1$  and  $\Psi_2$  provide respectively lower bounds for  $x_1$  and  $x_2$ , and these variables may be minimized out as

$$\begin{cases} x_1 = 0.0193x_3 \\ x_2 = 0.00954x_3 \end{cases} \quad (A1)$$

Substituting these relationships into the original problem, we have

$$\begin{aligned} \min \bar{\Psi}_0 &\equiv 0.01319166x_3^2x_4 + 0.024353275x_3^3 \\ \text{s.t.} \\ \bar{\Psi}_3 &\equiv 1296000 - \pi x_3^2x_4 - (4/3)\pi x_3^3 \leq 0 \\ 10 &\leq x_3, x_4 \leq 200 \end{aligned} \quad (A2)$$

where the upper bars on the cost and constraint symbols mean we are determining by now the solution for all variables continuous. The Lagrangian function for this problem is

$$L = \bar{\Psi}_0 + \lambda_3 \bar{\Psi}_3 - \mu_3^-(x_3 - 10) + \mu_3^+(x_3 - 200) - \mu_4^-(x_4 - 10) + \mu_4^+(x_4 - 200) \quad (A3)$$

where  $\lambda_3$ ,  $\mu_3^-$ ,  $\mu_3^+$ ,  $\mu_4^-$ ,  $\mu_4^+$  are the Lagrange multipliers for the constraint  $\bar{\Psi}_3$  and side constraints. The necessary Karush-Kuhn-Tucker conditions (Arora, 1989) for the problem (A.2) may now be set as

$$\begin{aligned} \partial L / \partial x_3 &\equiv 2 \times 0.01319166x_3x_4 + 3 \times 0.024353275x_3^2 \\ &\quad - 2\pi\lambda_3(x_3x_4 + 2x_3^2) - \mu_3^- + \mu_3^+ = 0 \\ \partial L / \partial x_4 &\equiv 0.01319166x_3^2 - \pi\lambda_3x_3^2 - \mu_4^- + \mu_4^+ = 0 \\ \lambda_3 \bar{\Psi}_3 &\equiv \lambda_3 [1296000 - \pi x_3^2x_4 - (4/3)\pi x_3^3] = 0 \\ \mu_k^-(x_k - 10) &= 0, \quad \mu_k^+(x_k - 200) = 0, \quad k = 3, 4 \\ \lambda_3, \mu_3^-, \mu_3^+, \mu_4^-, \mu_4^+ &\geq 0 \\ \bar{\Psi}_3 &\equiv 1296000 - \pi x_3^2x_4 - (4/3)\pi x_3^3 \leq 0 \\ 10 &\leq x_3, x_4 \leq 200 \end{aligned} \quad (A4)$$

Let us now search the different combinations of Lagrange multipliers:

1.  $\mu_3^- > 0, \mu_4^- > 0 \Rightarrow$   
 $x_3 = x_4 = 10, \Psi_3 = 1288669.6$  (violated)  
 $\mu_3^- > 0, \mu_4^+ > 0 \Rightarrow$   
 $x_3 = 10, x_4 = 200, \Psi_3 = 1228979.4$  (violated)  
 Then, whatever the value of  $\lambda_3$ , must have  $\mu_3^- = 0$

2.  $\lambda_3 = 0$ 
  - 2.1  $\mu_3^- = 0 \Rightarrow \mu_3^+ < 0$  (from the 1<sup>st</sup> condition, then violating the 5<sup>th</sup> one)
  - 2.2  $\mu_4^- = 0 \Rightarrow \mu_4^+ < 0$  (from the 2<sup>nd</sup> condition, then violating the 5<sup>th</sup> one)

3.  $\lambda_3 > 0 \Rightarrow \Psi_3 = 0 \therefore x_4 = 1296000 / (\pi x_3^2) - (4/3)x_3$ 
  - 3.1  $\mu_3^+ = \mu_4^- = \mu_4^+ = 0 \Rightarrow \lambda_3 = 0.01319166 / \pi$   
 (from the 2<sup>nd</sup> condition) and  $x_3 = 0$  (after substituting  $x_4$  and  $\lambda_4$  into the 1<sup>st</sup> one)
  - 3.2  $\mu_3^+ = \mu_4^- = 0, \mu_4^+ > 0 \Rightarrow x_4 = 200$  (from the 4<sup>th</sup> condition),

$$1296000 - \pi x_3^2 x_4 - (4/3)\pi x_3^3 = 0 \Rightarrow$$

$$x_3 = 40.3196187244$$

$$\lambda_3 = \frac{2 \times 0.01319166 x_3 x_4 + 3 \times 0.024353275 x_3^2}{2\pi(x_3 x_4 + 2x_3^2)}$$

$$= 0.004663057579 > 0$$

$$\mu_4^+ = \pi x_3^2 \lambda_3 - 0.01319166 x_3^2$$

$$= 2.369851195 > 0$$

All the optimality necessary conditions are satisfied, then

$$\begin{cases} \bar{x}_1 = 0.0193x_3 = 0.778168646 \\ \bar{x}_2 = 0.00954x_3 = 0.384649165 \\ \bar{x}_3 = 40.3196187244 \\ \bar{x}_4 = 200 \end{cases}$$

is candidate local optimum point for the assumed continuous variables.

- 3.3  $\mu_3^+ = \mu_4^+ = 0, \mu_4^- > 0 \Rightarrow x_4 = 10$  (from the 4<sup>th</sup> condition),

$$1296000 - \pi x_3^2 x_4 - (4/3)\pi x_3^3 = 0 \Rightarrow$$

$$x_3 = 65.2252326139$$

$$\lambda_3 = \frac{2 \times 0.01319166 x_3 x_4 + 3 \times 0.024353275 x_3^2}{2\pi(x_3 x_4 + 2x_3^2)}$$

$$= 0.005698937505 > 0$$

$$\mu_4^- = 0.01319166 x_3^2 - \pi x_3^2 \lambda_3$$

$$= -20.04674872 < 0$$

(violates 5<sup>th</sup> condition)

- 3.4  $\mu_4^- = \mu_4^+ = 0, \mu_3^+ > 0 \Rightarrow x_3 = 200 \Rightarrow$   
 $x_4 = -256.3534264 < 0$
- 3.5  $\mu_4^- = 0, \mu_3^+ > 0, \mu_4^+ > 0 \Rightarrow x_3 = x_4 = 200 \Rightarrow$   
 $\Psi_3 = -57347062.87 \neq 0$   
 (contradicts point 3:  $\Psi_3 = 0$ )
- 3.6  $\mu_4^+ = 0, \mu_3^+ > 0, \mu_4^- > 0 \Rightarrow x_3 = 200, x_4 = 10$   
 $\Rightarrow \Psi_3 = -33470958.7 \neq 0$   
 (contradicts point 3:  $\Psi_3 = 0$ )

Testing now the second-order sufficient conditions for the only point  $\bar{x} \equiv (\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4)$ , determined in 3.3, satisfying the necessary conditions, one may use the so-called bordered Hessian (Luenberger, 1984) calculated at that point:

$$B(\bar{x}_3, \bar{x}_4) \equiv \begin{bmatrix} 0 & \partial\Psi_3/\partial x_3 & \partial\Psi_3/\partial x_4 \\ \partial\Psi_3/\partial x_3 & \partial^2 L/\partial x_3^2 & \partial^2 L/\partial x_3 \partial x_4 \\ \partial\Psi_3/\partial x_4 & \partial^2 L/\partial x_3 \partial x_4 & \partial^2 L/\partial x_4^2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 & -71095.91969 & -5107.198124 \\ -71095.91969 & 232.2341915 & -0.117553254 \\ -5107.198124 & -0.117553254 & 0 \end{bmatrix}$$

As  $n - m$  for the problem (A.2) is  $2 - 1 = 1$  one has to calculate the last principal minor  $\det(B)$ . Since its value is negative, its sign is coincident with  $sign(-1)^m = sign(-1)$ . Hence, the Hessian of  $L$  is positive-definite and the point  $\bar{x}$  is a minimum point.

Now, rounding up the values of  $\bar{x}_1$  and  $\bar{x}_2$  to the table values, we have

$$\begin{cases} x_1^* = 0.8125 \\ x_2^* = 0.4375 \end{cases}$$

then determine the other two variables as

$$x_3^* = \min\{0.8125/0.0193, 0.4375/0.00954\}$$

$$= 42.0984456$$

From the two first constraints  $x_3 \leq 0.8125/0.0193$ ,  $x_3 \leq 0.4375/0.00954$ , i.e., the constraint  $\Psi_1$  is active at the optimum, and

$$x_4^* = 176.6365958$$

from the condition of  $\Psi_3 = 0 \therefore$

$$x_4 = 1296000 / (\pi x_3^2) - (4/3)x_3$$

Therefore, the analytical optimum point for the pressure vessel design problem is

$$x^* = (0.8125, 0.4375, 42.0984456, 176.6365958)$$

giving the minimum optimum cost  $\Psi_0^* = 6059.714335$ . At the optimum, the constraints have  $\Psi_1^* = \Psi_3^* = 0$  and  $\Psi_2^* = -0.082013323$ .

### Annex B: Tension-Compression Spring Classical Design Optimization

Again, let us firstly to analyze the monotonicity of the problem (Papalambros, Wilde, 1988). One should observe the constraint  $\Psi_1$  is critical with respect to the design variable  $x_3$ . The cost function  $\Psi_0$  increases monotonically in the variable  $x_3$ , and there is exactly one constraint, the constraint  $\Psi_1$ , whose monotonicity with respect to  $x_3$  is opposite from that of the objective. Then,  $\Psi_1$  is active and bounds  $x_3$  from below:

$$x_3 = 71785 x_1^4 / x_2^3 \tag{B1}$$

Substituting the relationship (B.1) into the objective function and into the constraint  $\Psi_3$  we get

$$\begin{cases} \Psi_0 = 2 x_1^2 x_2 + 71785 x_1^6 / x_2^2 \\ \Psi_3 = 1 - 0.001956536881 x_2 / x_1^3 \leq 0 \end{cases} \tag{B2}$$

Substituting the lower and upper bounds of  $x_2$  into the constraint  $\Psi_3$  we have that  $0.25 x_1 \leq 0.078790891$ ,  $1.3 x_1 \leq 0.136503503$ ; then the range of the design variable  $x_1$  can be set up as

$$0.05 \leq x_1 \leq 0.136503503$$

If one uses now the upper bounds of  $x_1$  and  $x_2$  into the constraint  $\Psi_4$ ,  $0.136503503 + 1.3 \leq 1.5$ , it is obvious that this constraint is always inactive, not playing any role into the optimization problem.

Studying now the monotonicity of the objective expressed in (B.2) with respect to the design variable  $x_2$ , the minimum of  $\Psi_0$  is given as

$$\begin{aligned} \partial \Psi_0 / \partial x_2 &\equiv 2 x_1^2 - 2 \times 71785 x_1^6 / x_2^3 = 0 \quad \therefore \\ x_2 &= \sqrt[3]{71785 x_1^4} \end{aligned}$$

since  $\partial^2 \Psi_0 / \partial x_2^2 \equiv 6 \times 71785 x_1^6 / x_2^4 > 0$ . Thus,  $\Psi_0$  decreases monotonically in  $x_2$  increase for  $0.25 \leq x_2 \leq \sqrt[3]{71785 x_1^4}$  and increases monotonically in  $x_2$  increase for  $\sqrt[3]{71785 x_1^4} \leq x_2 \leq 1.3$ . For example, within the range  $0.25 \leq x_2 \leq \sqrt[3]{71785 x_1^4}$ , the function  $\Psi_0$  decreases in the variable  $x_2$  increase, achieving the minimum at  $x_2 = 0.765545910$  for a prescribed  $x_1 = 0.05$ , and increases the value of the minimum point at  $x_2$  as  $x_1$  increases. For  $x_1 \geq 0.074378786$  the function

$\Psi_0$  is a decreasing function all along the feasible domain of  $x_2$ ,  $0.25 \leq x_2 \leq 1.3$ .

The constraint  $\Psi_2$  may be expressed as

$$\begin{aligned} 4 x_2^2 + (C - 1) x_1 x_2 - C x_1 &\leq 0, \\ C &= 2.460062647 - 12566 x_1^2 \end{aligned}$$

This constraint increases monotonically in  $x_2$  increase; then its monotonicity with respect to  $x_2$  is opposite from that of  $\Psi_0$  for  $0.25 \leq x_2 \leq \sqrt[3]{71785 x_1^4}$  and the constraint  $\Psi_2$  is critical providing an upper bound for  $x_2$ :

$$x_2 = (x_1 / 8) \left( 1 - C + \sqrt{C^2 + 14C + 1} \right)$$

The reduced problem may now be expressed as

$$\begin{aligned} \min \Psi_0 &\equiv 2 x_1^2 x_2 + 71785 x_1^6 / x_2^2 \\ \text{s.t.} & \\ x_2 &= (x_1 / 8) \left( 1 - C + \sqrt{C^2 + 14C + 1} \right) \\ 0.05 \leq x_1 &\leq 0.136503503, \quad 0.25 \leq x_2 \leq 1.3 \end{aligned} \tag{B3}$$

The optimum point can be determined easily by using a unidimensional search in  $x_1$ , with the active constraint  $\Psi_2$  determining  $x_2$ . The variable  $x_3$  is calculated by using the relationship (B.1) after solving the problem (B.3).

The optimum value of the objective function is obtained as  $\Psi_0^* = 0.012665232$  at the point  $x^* \equiv (0.051690, 0.356740328, 11.28764160)$ . At the optimum, the constraints have the values  $\Psi_1^* = 0$ ,  $\Psi_2^* = 0$ ,  $\Psi_3^* = -4.05383024$ ,  $\Psi_4^* = -0.727713114$ .



# The Iris Dataset Revisited – a Partial Ordering Study

Lars Carlsen

Awareness Center, Linkøpingvej 35, Trekroner, DK-4000 Roskilde4, Denmark  
E-mail: LC@AwarenessCenter.dk

Rainer Bruggemann

Leibniz-Institute of Freshwater Ecology and Inland Fisheries, Department: Ecohydrology  
D-92421 Schwandorf, Oskar - Kösters-Str. 11, Germany  
E-mail: brg\_home@web.de

**Keywords:** IRIS data set, partial ordering, separability, dominance, classification

**Received:** March 11, 2019

*The well-known Iris data set has been studied applying partial ordering methodology. Previous studies, e.g., applying supervision learning such as neural networks (NN) and support-vector machines (SVM) perfectly distinguish between the three Iris subgroups, i.e., Iris Setosa, Iris Versicolour and Iris Virginica, respectively, in contrast to, e.g., K-means clustering that only separates the full Iris data set in two clusters. In the present study applying partial ordering methodology further discloses the difference between the different classification methods. The partial ordering results appears to be in perfect agreement with the results of the K-means clustering, which means that the clear separation in the three Iris subsets applying NN and SVM is neither recognized by clustering nor by partial ordering methodology.*

*Povzetek: Analizirana je znana baza učnih domen Iris s poudarkom na nekaterih metodah, recimo gručenju.*

## 1 Introduction

One of the most often applied datasets in machine learning studies test cases is the Iris dataset [1, 2]. This dataset includes 150 entries comprising 3 x 50 entries for three subspecies of class of iris plant, i.e., Iris Setosa (i-set), Iris Versicolour (i-ver) and Iris Virginica (i-vir), respectively. The plants are characterized by four indicators, i.e., Sepal length (SepalL), Sepal width (SepalW), Petal length (PetalL) and Petal width (PetalW), respectively, all in cm.

We find that supervised learning, like neural network and SVM, nicely classify the 3 classes Iris Setosa (i-set), Iris Versicolour (i-ver) and Iris Virginica (i-vir). Using 60% randomly chosen entries as test set a neural network with one hidden layer with 3 nodes leads to only one misclassification between the remaining 40% of the entries serving as validation set. A similar result was obtained using a SVM approach with a radial kernel. Here we find two misclassifications of i-vir being classified as i-ver. In the case of K-means clustering a somewhat less clear picture develops.

In a recent chemistry-oriented study we investigated the potential use of partial ordering methodology as a tool for classification of alkyl anilines [3]. In the present study we apply partial order methodologies to further study as to how far the supervised classification of the three types of Iris-plants can be re-found.

The mathematical theory of partial orders seems to be started in the late 19<sup>th</sup> century (cf. [4]), however, the main development to establish an own mathematical

discipline with the methodological components of combinatorics, algebra and graph theory, can be attributed to the work of Birkhoff [5] and Hasse [6]. To our knowledge there were only few applications of the theory of partial order, i.e., in statistics, e.g. [7-9], concepts of phase transfers [10] and early electronics [11]. In chemistry important and theoretically attractive applications were found by Ruch [12]. Nevertheless, these concepts became not popular, albeit their theoretical beauty. With publications of Randić [13] and Klein [14] and after the pioneering work of Halfon and Reggiani [15] the mathematical theory of partial order became a useful tool in environmental sciences. The background of this development is that environmental systems are complex and a decision about environmental issues had to be based on a set of indicators, describing the state of the environmental system. However, decisions based on a set of indicators are difficult and caused the usage of multicriteria decision aids (MCDA). Famous MCDA-methods are PROMETHEE [16], Electre [17] and partial order concepts [18]. Today the partial order theory is further developed, mainly in the field of multi indicator-systems and is recently applied in social sciences too, e.g. [19]. The latest methodological development is mainly focused on, how to include data uncertainty [20].

## 2 Materials and methods

### 2.1 Data

The data for the current study is the well-known Iris dataset [1] comprising 3 x 50 entries for three subspecies of class of iris plant, i.e., Iris Setosa (i-set), Iris Versicolour (i-ver) and Iris Virginica (i-vir), respectively. Thus, the Iris dataset comprises in total 150 entries.

### 2.2 Basic concepts of partial order

Let  $X$  be a set of objects, labeled by  $x_i$  ( $i = 1, \dots, n$ ), which can be for example chemical compounds. To define an order relation among them, the relation “ $\leq$ ” has to obey the following order axioms:

- reflexivity: the object can be compared with itself, i.e.,  $x \leq x$
- antisymmetry: if  $x \leq y$  and  $y \leq x \Rightarrow x = y$
- transitivity: if  $x \leq y$  and  $y \leq z \Rightarrow x \leq z$

A special realization of order relations is given by eq. 1. Equation 1 expresses a mapping from object  $x$  to its representation by a tuple  $q$  with  $m$  components, as well as the order relation among objects defined by the simultaneous evolution of the tuple  $q$ .

- $x \rightarrow q$  (the set of objects,  $X$ , is mapped onto the set of tuples  $\{q\}$ , by assigning to each object  $x$  its tuple, based on the values of the considered indicators, i.e.,
- $x \rightarrow (q_1(x), q_2(x), \dots, q_m(x))$  where  $q_j$  ( $j=1, \dots, m$ ) is a selection of certain properties of  $x$
- $x \leq y : \Leftrightarrow (q_j(x) \leq q_j(y) \text{ for all } j$  (1)

By eq. 1 a partial order is defined, and the object set  $X$  is by eq. 1 equipped with partial order relation; such a set is called a partially ordered set, in brief poset, denoted often as:  $(X, \leq)$ . If neither  $x \leq y$  nor  $y \leq x$ ,  $x$  is said to be incomparable with  $y$ , denoted as  $x \parallel y$ . By eq. 1 an order relation is defined, if  $x \leq y$  or  $y \leq x$ . The presence of an order relation can be described by the zeta matrix. The zeta matrix is defined as follows:

$$z_{i,j} = \begin{cases} 1 & \text{if } x_i \leq x_j \\ 0 & \text{otherwise} \end{cases}$$

From transitivity it follows that in case of  $x \leq z$  and  $z \leq y$  the implication  $x \leq y$  can be deduced from the premise, i.e., a rational description of the partial order

can be given by the cover relation: “ $y$  is covering  $x$ , if  $x < y$  and there is no other element  $z$ , for which  $x < z$  and  $z \leq y$ . Both relations, i.e., the order and the cover-relation can be expressed by adjacency like matrices.

By application of the cover-relation a graph is constructed. This graph is, based on the three axioms of partial order

- directed (due to the order relation)
- triangle free (due to the cover relation) and
- does not contain cycles, due to the antisymmetry.

By convention, originally introduced by Halfon and Reggiani [15], the graph is drawn with

- $x \leq y$  locating  $x$  below  $y$ ,
- attempting a symmetric presentation as far as possible
- by an arrangement of objects in levels.

For a detailed explanation see [18]. For examples, see sect. 3.

### 2.3 Levels

A poset  $(X, \leq)$  can be partitioned into a family of subsets  $X_i \subset X$ :

$$(X, \leq) = \oplus (X_i, \leq), \text{ i.e. } X_i, X_j \subset X \text{ and } X_i \cap X_j = \emptyset, i \neq j \quad (2)$$

The symbol  $\oplus$  is a shorthand notation for the union of mutually non-intersecting subsets whereby any pair  $x, y$  with  $x \in X_i$  and  $y \in X_j$  (with  $i \neq j$ ) implies  $x \parallel y$ .

The family of sets  $X_i$  can be ordered, i.e.,  $X_{i1} < X_{i2}$ :  $\Leftrightarrow$  there is an element in  $X_{i1}$  which is covered by an element in  $X_{i2}$ . The sets  $X_i$  obeying the above relations are called levels.

The dissection of  $X$  into levels, i.e., into subsets obeying not only the order theoretical characterization, given above, but also eq. 2, allows a geometrical representation by a so-called Hasse diagram that can be seen as a rectangle, filled with the bottom level, then the next level, until the top level is reached. Important is the possibility to perform for each level a permutation (this is possible because there is no order relation among the elements of a level) in that manner, that supervised subsets can be given specific locations within any level. If for example a Hasse diagram has four levels, then its representation by level may look like that in Fig. 1.

Table 1.: i-set/i-ver, i-set/i-vir and i-ver/i-vir ratios for the 50 sample in the three Iris set set1, set2 and set3.

No	set1 (i-set)				set2 (i-ver)				set3 (i-vir)			
	SepalL	SepalW	PetalL	PetalW	SepalL	SepalW	PetalL	PetalW	SepalL	SepalW	PetalL	PetalW
1	0.73	1.09	0.30	0.14	0.81	1.06	0.23	0.08	1.11	0.97	0.78	0.56
2	0.77	0.94	0.31	0.13	0.84	1.11	0.27	0.11	1.10	1.19	0.88	0.79
3	0.68	1.03	0.27	0.13	0.66	1.07	0.22	0.10	0.97	1.03	0.83	0.71
4	0.84	1.35	0.38	0.15	0.73	1.07	0.27	0.11	0.87	0.79	0.71	0.72
5	0.77	1.29	0.30	0.13	0.77	1.20	0.24	0.09	1.00	0.93	0.79	0.68

6	0.95	1.39	0.38	0.31	0.71	1.30	0.26	0.19	0.75	0.93	0.68	0.62
7	0.73	1.03	0.30	0.19	0.94	1.36	0.31	0.18	1.29	1.32	1.04	0.94
8	1.02	1.42	0.45	0.20	0.68	1.17	0.24	0.11	0.67	0.83	0.52	0.56
9	0.67	1.00	0.30	0.15	0.66	1.16	0.24	0.11	0.99	1.16	0.79	0.72
10	0.94	1.15	0.38	0.07	0.68	0.86	0.25	0.04	0.72	0.75	0.64	0.56
11	1.08	1.85	0.43	0.20	0.83	1.16	0.29	0.10	0.77	0.63	0.69	0.50
12	0.81	1.13	0.38	0.13	0.75	1.26	0.30	0.11	0.92	1.11	0.79	0.79
13	0.80	1.36	0.35	0.10	0.71	1.00	0.25	0.05	0.88	0.73	0.73	0.48
14	0.70	1.03	0.23	0.07	0.75	1.20	0.22	0.05	1.07	1.16	0.94	0.70
15	1.04	1.38	0.33	0.15	1.00	1.43	0.24	0.08	0.97	1.04	0.71	0.54
16	0.85	1.42	0.34	0.29	0.89	1.38	0.28	0.17	1.05	0.97	0.83	0.61
17	0.96	1.30	0.29	0.27	0.83	1.30	0.24	0.22	0.86	1.00	0.82	0.83
18	0.88	1.30	0.34	0.30	0.66	0.92	0.21	0.14	0.75	0.71	0.61	0.45
19	0.92	1.73	0.38	0.20	0.74	1.46	0.25	0.13	0.81	0.85	0.65	0.65
20	0.91	1.52	0.38	0.27	0.85	1.73	0.30	0.20	0.93	1.14	0.78	0.73
21	0.92	1.06	0.35	0.11	0.78	1.06	0.30	0.09	0.86	1.00	0.84	0.78
22	0.84	1.32	0.38	0.31	0.91	1.32	0.31	0.20	1.09	1.00	0.82	0.65
23	0.73	1.44	0.20	0.13	0.60	1.29	0.15	0.10	0.82	0.89	0.73	0.75
24	0.84	1.18	0.36	0.42	0.81	1.22	0.35	0.28	0.97	1.04	0.96	0.67
25	0.75	1.17	0.44	0.15	0.72	1.03	0.33	0.10	0.96	0.88	0.75	0.62
26	0.76	1.00	0.36	0.14	0.69	0.94	0.27	0.11	0.92	0.94	0.73	0.78
27	0.74	1.21	0.33	0.29	0.81	1.21	0.33	0.22	1.10	1.00	1.00	0.78
28	0.78	1.17	0.30	0.12	0.85	1.17	0.31	0.11	1.10	1.00	1.02	0.94
29	0.87	1.17	0.31	0.13	0.81	1.21	0.25	0.10	0.94	1.04	0.80	0.71
30	0.82	1.23	0.46	0.20	0.65	1.07	0.28	0.13	0.79	0.87	0.60	0.63
31	0.87	1.29	0.42	0.18	0.65	1.11	0.26	0.11	0.74	0.86	0.62	0.58
32	0.98	1.42	0.41	0.40	0.68	0.89	0.23	0.20	0.70	0.63	0.58	0.50
33	0.90	1.52	0.38	0.08	0.81	1.46	0.27	0.05	0.91	0.96	0.70	0.55
34	0.92	1.56	0.27	0.13	0.87	1.50	0.27	0.13	0.95	0.96	1.00	1.07
35	0.91	1.03	0.33	0.13	0.80	1.19	0.27	0.14	0.89	1.15	0.80	1.07
36	0.83	0.94	0.27	0.13	0.65	1.07	0.20	0.09	0.78	1.13	0.74	0.70
37	0.82	1.13	0.28	0.13	0.87	1.03	0.23	0.08	1.06	0.91	0.84	0.63
38	0.78	1.57	0.32	0.08	0.77	1.16	0.25	0.06	0.98	0.74	0.80	0.72
39	0.79	1.00	0.32	0.15	0.73	1.00	0.27	0.11	0.93	1.00	0.85	0.72
40	0.93	1.36	0.38	0.15	0.74	1.10	0.28	0.10	0.80	0.81	0.74	0.62
41	0.91	1.35	0.30	0.25	0.75	1.13	0.23	0.13	0.82	0.84	0.79	0.50
42	0.74	0.77	0.28	0.21	0.65	0.74	0.25	0.13	0.88	0.97	0.90	0.61
43	0.76	1.23	0.33	0.17	0.76	1.19	0.25	0.11	1.00	0.96	0.78	0.63
44	1.00	1.52	0.48	0.60	0.74	1.09	0.27	0.26	0.74	0.72	0.56	0.43
45	0.91	1.41	0.45	0.31	0.76	1.15	0.33	0.16	0.84	0.82	0.74	0.52
46	0.84	1.00	0.33	0.25	0.72	1.00	0.27	0.13	0.85	1.00	0.81	0.52
47	0.89	1.31	0.38	0.15	0.81	1.52	0.32	0.11	0.90	1.16	0.84	0.68
48	0.74	1.10	0.33	0.15	0.71	1.07	0.27	0.10	0.95	0.97	0.83	0.65
49	1.04	1.48	0.50	0.18	0.85	1.09	0.28	0.09	0.82	0.74	0.56	0.48
50	0.88	1.18	0.34	0.15	0.85	1.10	0.27	0.11	0.97	0.93	0.80	0.72

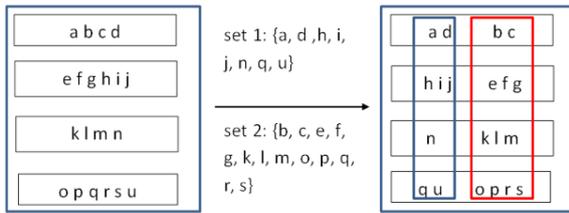


Figure 1: Within any level the position of objects can be freely permuted, nevertheless keeping their original order relations.

### 2.4 Dominance and separability

A partially ordered set may be partitioned in any other manner, i.e., not following the level-construction, but following a supervised classification, e.g., through an aggregation process [21]. Let  $X$  be partitioned into a family of sets  $X_i$ ,  $i = 1, \dots, r$ , where the sets  $X_i$  are externally defined. Then the natural question is, as to how far the family of sets  $\{X_i\}$  can be partially ordered. This question was in depth analyzed by Restrepo and Bruggemann [22]. Here, however, we follow a different concept.

Assume that by any cluster analysis (K-means, hierarchical models) the family of subsets  $X_i$  is defined. What at best can be expected from an analysis by partial order? In order to arrive at this aim, the partial order concepts of linear sums and disjoint union of sets must be defined:

Let  $X_i, X_j$  be subsets of  $X$ , then:

$X_i$  and  $X_j$  form a *linear sum*, denoted as  $X_i \oplus_{\leq} X_j$  (in order to differentiate this symbol from that in eq. 2, we add as a subscript “ $\leq$ ”) when:

For all  $x \in X_i$ , and all  $y \in X_j$   $x \leq y$ ,

**Similarly:**

$X_i$  and  $X_j$  form a *complete disjoint union of sets*, denoted as  $X_i \cup_{\leq} X_j$  (in order to differentiate this symbol from the union symbol in set theory, we add as a subscript “ $\leq$ ”) when:

For all  $x \in X_i$ , and all  $y \in X_j$   $x \parallel y$ ,

A pretty clear classification by partial order theory can be obtained, when the poset is either a linear sum, cf. [21] or a complete disjoint union of sets [23]. Such a classification due to partial order concepts can be visualized as shown in Fig. 2.

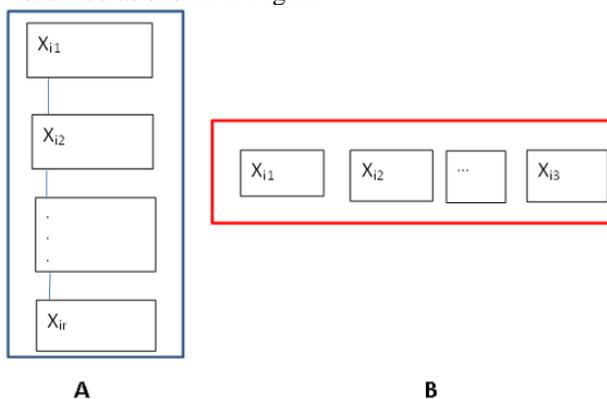


Figure 2: Two extremal cases for a poset representation of a supervised classification (see text).

In Fig. 2A  $x \geq y$  for any  $x \in X_{ik}$  and  $y \in X_{ik+1}$ . A partially ordered set representable by Fig. 2A is called a linear sum. In contrast to the linear sum construction is the disjoint union of sets  $X_{ik}$  in Fig. 2B. Here the following is valid:  $x \parallel y$  for all  $x \in X_{ik}$  and  $y \in X_{ik+s}$ .

Structures like those shown in Fig. 2 cannot be expected within a real data set. The question is, as to how far approximations corresponding to the two archetypes of Fig. 2 can be found. Hereto two matrices, dominance and separability are introduced.

The dominance matrix being defined as follows

$$\text{Dom}(X_i, X_j) = |\{(x, y) \text{ with } x \in X_i \text{ and } y \in X_j \text{ and } x > y\}| / |X_i| * |X_j|, \text{ dominance matrix} \quad (3)$$

In a situation as shown in Fig. 3a,  $\text{Dom}(X_{ik}, X_{ik+s}) = 1$ , in all other cases  $0 \leq \text{Dom}(X_i, X_j) < 1$ . We speak of  $x_i$  as approximately dominating  $x_j$ , if  $\text{Dom}(X_i, X_j) > 0.5$ .

The analog to Fig. 2B is the separability matrix defined as

$$\text{Sep}(X_i, X_j) = |\{(x, y) \text{ with } x \in X_i \text{ and } y \in X_j \text{ and } x \parallel y\}| / |X_i| * |X_j|, \quad (4)$$

A situation such as shown in Fig. 2B would lead to  $\text{Sep}(X_i, X_j) = 1$ , in cases  $\text{Sep}(X_i, X_j) > 0.5$  we speak of an approximation with respect to Fig. 2B.

### 2.5 Software

The applied software is PyHasse, programmed applying the programming (interpreter) language Python and, in honor of Helmut Hasse, who was one of the main mathematicians, investigating partial order. The complete PyHasse software package is available from Dr. Bruggemann (brg\_home@web.de). A limited version can be assessed at [www.pyhasse.org](http://www.pyhasse.org) (For further details, see [24, 25]).

## 3 Results and discussion

### 3.1 K-means clustering

K-means clustering and Hierarchical clustering (HCA) apparently does lead to less clear-cut pictures. Thus, we find that K-means clustering virtually separated i-set from i-vir and i-ver, whereas a separation of i-vir and i-ver is significantly less pronounced (Fig. 3A). This is in agreement with the plots shown in [2]. A further analyses including the i-ver and i-vir sets lead to some separation between the two sets although a significant overlap is seen (Fig. 3B).

The answer to the somewhat surprising K-means clustering can be found in the data shown in Table 1. Table 1 discloses the i-set/i-ver, i-set/i-vir and i-ver/i-vir ratios for the 50 sample in the three iris sets set 1, set2 and set 3. It is immediate noted that in the cases of the i-set/i-ver and i-set/i-vir the ratios are significantly different from 1, whereas in the case of i-ver/i-vir th ratio values are in most case rather close to 1, explaining the lack of separation between), Iris Versicolour and Iris Virginica as displayed in the K-means clustering.

A further discussion of the K-means clustering that is a well-established method is not in the focus of the present paper.

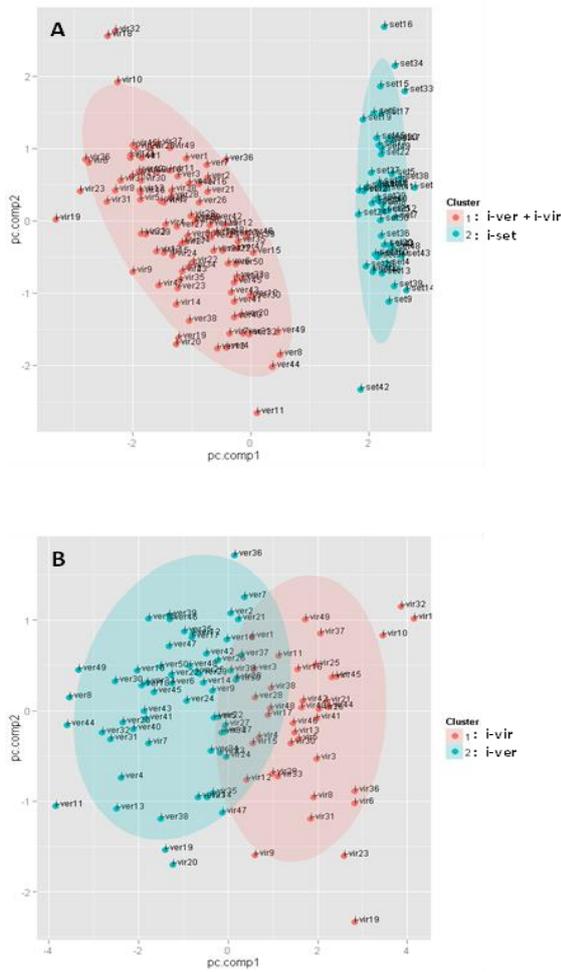


Figure 3: K-means clustering of A: the complete Iris data set and B: the i-ver and i-vir sets

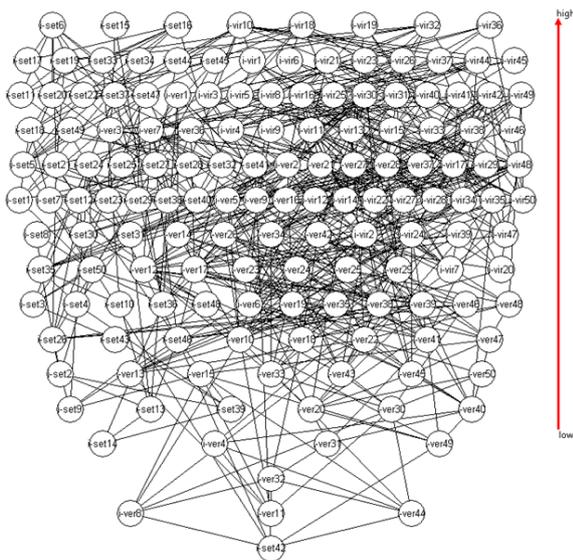


Figure 4: The Hasse diagram of complete Iris set under the four indicators. The diagram displays 4150 comparabilities and 6876 incomparabilities.

### 3.2 The Hasse diagram - visual inspection

The Hasse diagram of the complete Iris dataset is found in Fig. 4.

Inspecting Fig. 4 a substructure, which hopefully mimics the three Iris-families, can obviously not be recognized. There is no clear separation in the sense of eq. 4, which can be just visually detected.

The tools outlined in sect. 2.4 may be helpful to find a structure in the Hasse diagram, when the classification into the three Iris-subsets is used. Hence, we sharpen our message to: Given the classification into the three Iris-subsets, what posetic relations among these three subsets can be found. As visual techniques fail, numerical devices as the dominance and separability matrices are necessary.

### 3.3 Supervised classification

#### 3.3.1 The subsets

The complete Iris data set is subsetted into three sets comprising the 3 Iris species Iris Setosa (i-set), Iris Versicolour (i-ver) and Iris Virginica (i-vir), respectively. Hence,

set 1:  
 i-set1, i-set2, i-set3, i-set4, i-set5, i-set6, i-set7, i-set8, i-set9, i-set10, i-set11, i-set12, i-set13, i-set14, i-set15, i-set16, i-set17, i-set18, i-set19, i-set20, i-set21, i-set22, i-set23, i-set24, i-set25, i-set26, i-set27, i-set28, i-set29, i-set30, i-set31, i-set32, i-set33, i-set34, i-set35, i-set36, i-set37, i-set38, i-set39, i-set40, i-set41, i-set42, i-set43, i-set44, i-set45, i-set46, i-set47, i-set48, i-set49, i-set50

set 2:  
 i-ver1, i-ver2, i-ver3, i-ver4, i-ver5, i-ver6, i-ver7, i-ver8, i-ver9, i-ver10, i-ver11, i-ver12, i-ver13, i-ver14, i-ver15, i-ver16, i-ver17, i-ver18, i-ver19, i-ver20, i-ver21, i-ver22, i-ver23, i-ver24, i-ver25, i-ver26, i-ver27, i-ver28, i-ver29, i-ver30, i-ver31, i-ver32, i-ver33, i-ver34, i-ver35, i-ver36, i-ver37, i-ver38, i-ver39, i-ver40, i-ver41, i-ver42, i-ver43, i-ver44, i-ver45, i-ver46, i-ver47, i-ver48, i-ver49, i-ver50

set 3:  
 i-vir1, i-vir2, i-vir3, i-vir4, i-vir5, i-vir6, i-vir7, i-vir8, i-vir9, i-vir10, i-vir11, i-vir12, i-vir13, i-vir14, i-vir15, i-vir16, i-vir17, i-vir18, i-vir19, i-vir20, i-vir21, i-vir22, i-vir23, i-vir24, i-vir25, i-vir26, i-vir27, i-vir28, i-vir29, i-vir30, i-vir31, i-vir32, i-vir33, i-vir34, i-vir35, i-vir36, i-vir37, i-vir38, i-vir39, i-vir40, i-vir41, i-vir42, i-vir43, i-vir44, i-vir45, i-vir46, i-vir47, i-vir48, i-vir49, i-vir50

#### 3.3.2 Dominances and separabilities

Applying the appropriate PyHasse module, mainly ddssimpl and the new module ddssimpl\_batch (for ddssimpl, cf. [22] the dominance and separability matrices for the three subsets are

Dominance matrix Dom ( $DOM(i,j)/(n_i*n_j)$ )

	1	2	3
1	0.252	0.0	0.0
2	0.094	0.31	0.002
3	0.196	0.63	0.256

Separability matrix Sep ( $SEP(i,j)/(n_i*n_j)$ )

	1	2	3
1	0.517	0.906	0.804
2	0.906	0.401	0.368
3	0.804	0.368	0.509

The separability matrix shows a clear separation between set1 and sets2 and set3, respectively, However, between set2 and set3 a considerable overlapping can be noted expressed by the relatively low value of the nondiagonal element  $Sep(2,3) = 0.368$ , which does not justify a separation in the sense of Fig. 3B. The dominance matrix shows correspondingly for the entry  $Dom(3,2)$  a value  $> 0.5$ . This result is in perfect agreement with the above discussed K-means results.

### 3.4 Separability matrix as mean to visualize the classification

Taken just the two sets, set1 (i-set) and set2 (i-ver), the

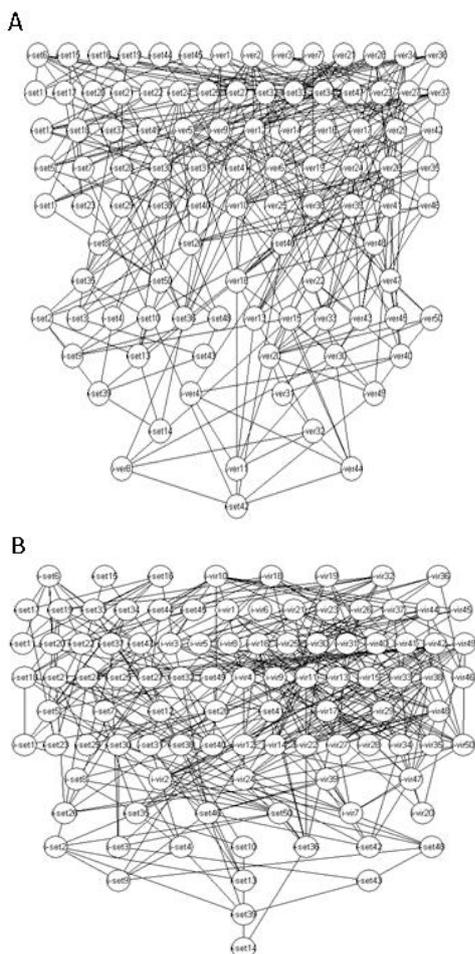


Figure 5: Hasse diagram of A: the i-set (set1) and i-ver (set2) and B: the i-set (set1) and i-vir (set3) under the four indicators.

Hasse diagram is shown in Fig. 5.

In contrast to the pretty clear separation between set1 and set2 (Table 5A) on the one hand, and set1 and set3 (Fig. 5B) on the other hand, the Hasse diagram, based on set2 and set3 only, shows a structure, which schematically could be visualized, as shown in Fig. 6.

The blue part is located below the orange colored part. Hence, there are many order relations of  $x \in set3$  and  $y \in set2$ , where  $x > y$  in the order-theoretical sense. This explains pictorially that  $dom(3,2) \gg dom(2,3)$  (vide supra). As the blue part is located on the left side, whereas the orange part of the right side of the schematic representation of a Hasse diagram, there are also many relations with  $x \parallel y$ . The order theory does not support a separation between set2 and set3. An enhancement with respect to dominance relations may be given by the two little rectangles in the middle of the scheme (Fig. 6), which we call a “nose”. Note that from a geometrical point of view, the elements of the “nose” could be arranged so that formally the two triangles are not perturbed. However, the “nose” indicates the count of elements which lead to the irregular structure, shown in Fig. 6. The real Hasse diagram of set2 and set3 is shown in Fig. 7, where the above schematic structure (Fig. 6) easily is recognized.

It remains to discuss two points:

1. What is the effect of the elements within the “nose” (section 3.4)?
2. What can be said about an internal partitioning of the sets i-set, i-ver and i-vir (section 3.5)?

### 3.5 Effect of the elements of the "nose"

In this section we investigate, as to how far elements of a specific geometric configuration, here the “nose” can influence the values of the dominance and separability matrix. It is to be clarified, whether or not such geometrical configuration destabilizes the conclusions based on the two concepts, i.e., dominance and



Figure 6: Schematic view, based on the methods, explained in sect. 2 for set2 and 3. The parts, obviously not being confined by the triangles are called “nose”.

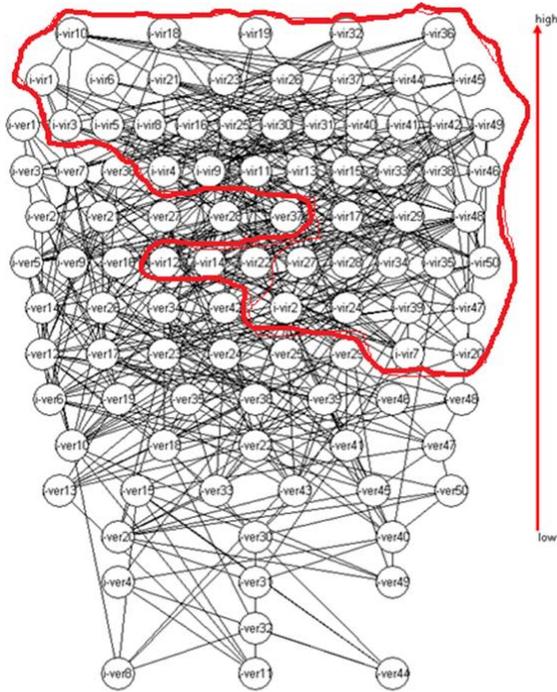


Figure 7: Hasse diagram of only set2 and set3. The separation between set3 and set2 is marked with a red line. The characteristic S-like curve marks the “noses”.

separability.

The elements of the “noses” are (cf. Fig. 7):

- i-ver: 27, 28, 37
- i-vir: 12, 14, 22, 27

There will be 7 runs:

- a. Keeping i-vir constant and eliminate one after another 27, 28, 37 to study the effect of the “noses” of i-ver and
- b. Keeping i-ver constant and eliminate 12, 14, 22, 27, respectively, one after another.

The results are shown in Table 2.

All in all the entries of the dominance and separability matrix are only slightly changed. The elements of the “nose” do not contribute much to the general dominance behavior, i.e., the presence of the “noses” does not change the impression that i-vir dominates to some degree the set i-ver. The separability values are in comparison to the standard reduced, showing that the elements of the “nose” contribute somewhat to the incomparabilities between the elements of i-vir and i-ver, respectively. When all elements of the “nose” are eliminated (9th row in Table 2) then the dominance of i-vir over i-ver is slightly enhanced, that of i-ver over i-vir reduced and the separation reduced. The comparison to the effect of the elements of the top level, namely i-vir10, i-vir18, i-vir19, i-vir32, i-vir36 shows that the position of elements to be eliminated within the level-order-system as schematically shown in Fig. 6 does not play a strong role.

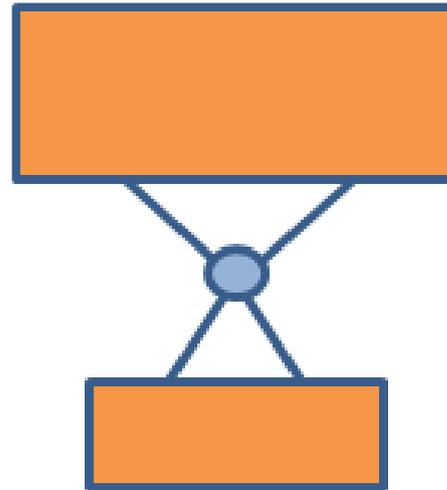


Figure 8: Scheme of the Hasse diagram, based on set1.

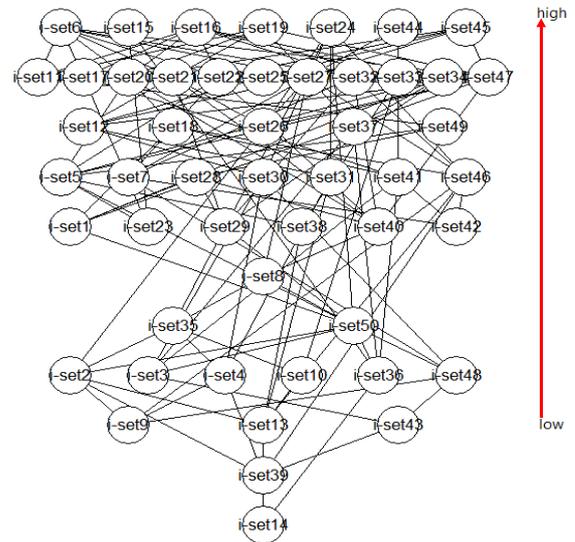


Figure 9: Hasse diagram of the i-set (set1). Total incomparabilities 646; Total comparabilities 579.

### 3.6 Internal structures

So far it has been demonstrated that posetic relations can roughly, but not convincingly verify the supervised classification. A further question remains, which is a partial-order-like point of view. Thus, do internal separations occur? Are there subsets of set1, set2 and set3, respectively that may be identified by partial order theory?

We answer these questions by an analysis of the i-set (set1). In Fig. 8 the Hasse diagram of set1 is shown.

By a simple optical inspection it is clear that there are two subsets which dominate each other to a striking degree. This situation can be schematically illustrated by Fig. 9.

In order not solely to rely on the visual impression, the dominance and separability approach (cf. sect. 3.2) is brought into play.

subsets	Dom(1,2) Dominance i-ver over i-vir	Dom(2,1) Dominance i-vir over i-ver	Sep(1,2) = Sep(2,1)
No. of elimination at all: Standard	0.002	0.637	0.362
Elim 27 from i-ver	0.002	0.643	0.355
Elim 28 from i-ver	0.001	0.643	0.356
Elim 37 from i-ver	0.002	0.645	0.353
Elim 12 from i-vir	0.002	0.641	0.357
Elim 14 from i-vir	0.002	0.638	0.361
Elim 22 from i-vir	0.002	0.645	0.353
Elim 27 from i-vir	0.002	0.641	0.358
i-ver without "nose"	0.001	0.687	0.312
I-vir without top level element 10 in comparison with i-ver	0.002	0.629	0.369
...Without 18	0.002	0.629	0.369
...Without 19	0.002	0.643	0.355
...Without 32	0.002	0.629	0.369
...Without 36	0.002	0.633	0.366

Table 2: Elimination of single elements from the “nose”.

### 3.6.1 Subset-selection:

In the following the 3 subsets, selected based on a visual inspection to be included in the analysis is shown. It should be noted that only representatives is shown, i.e., equivalent object being represented by one element only.

subset 1:

i-set6, i-set15, i-set16, i-set19, i-set24, i-set44, i-set45, i-set11, i-set17, i-set20, i-set21, i-set22, i-set25, i-set27, i-set32, i-set33, i-set34, i-set47, i-set12, i-set18, i-set26, i-set37, i-set49, i-set5, i-set7, i-set28, i-set30, i-set31, i-set41, i-set46, i-set1, i-set23, i-set29, i-set38, i-set40, i-set42

subset 2:

i-set8

subset 3:

i-set35, i-set50, i-set2, i-set3, i-set4, i-set10, i-set36, i-set48, i-set9, i-set13, i-set43, i-set39, i-set14,

### 3.6.2 Dominances and separabilities

The (relative) dominance matrix was calculated to be

Dominance matrix Dom (DOM(i,j)/(ni\*nj))

	1	2	3
1	0.176	0.417	0.688
2	0.0	1.0	1.0
3	0.0	0.0	0.296

The values of this matrix imply that there is a vague dominance of subset1 over subset2 (which is a singleton (namely comprising only i-set8), and a slightly clearer dominance of subset1 over subset3. It is further disclosed that i-set8 is higher located than all elements of subset3, i.e., completely dominating subset3.

Relative separability matrix was found to be

Separability matrix Sep (SEP(i,j)/(ni\*nj))

	1	2	3
1	0.676	0.583	0.312
2	0.583	0.0	0.0
3	0.312	0.0	0.485

Here it becomes clear that the incomparability among elements of subset1 and subset2 is the main result. Whereas the role of incomparabilities among elements of subset1 and those of subset3 is relatively low. Thus, a model for a classification for the i-set- series is the Hasse diagram, found in Fig. 10 using the PyHasse module ddssimpl1.py.

It can be seen, that due to the values of the dominance and separability matrices the differentiation between subset1 and subset3 is more pronounced than a differentiation within subset1.

Just by a simple visual inspection of Fig. 7 it is clear that a differentiation between i-vir and i-ver does not appear meaningful.

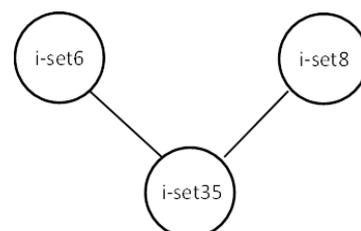


Figure 10: The representative elements of each subset are selected as labels for the Hasse diagram. The basic set is the i-set.



Figure 11: Schematic illustration of the separations of the 3 sets. Set1 is separated from set2 and set3, whereas the sets 2 and 3 have an overlapping zone.

## 4 Conclusions and outlook

The results of partial order theory suggest a pictorial representation as follows (Fig. 11)

The result summarized in Figure 11 is in perfect agreement with the results of the K-means clustering (cf. Fig. 1). Hence, the clear separation in the three sets is disclosed through the supervised learning by NN and SVM cannot be verified neither by clustering nor by partial ordering methodology.

The slight dominance of Set3 over Set2 is not indicated, it is discussed in more detail in Sect. 3. Correspondingly the internal structure within Set1 is not represented.

Section 3.4 shows that in the case of so many elements in each set (i-set, i-ver, i-vir), the effect of elements in the "nose" seems to play a minor role. These elements contribute slightly to incomparabilities, but do not change the dominance values. The reason is that these elements are mainly "in-between"-elements. Elements above the "nose" (i-vir) are still connected with elements below the "nose" (i-ver), independently whether elements in the "nose" are eliminated or not.

Section 3.5 shows, based on optical inspection of the corresponding Hasse diagrams that at maximum the i-set could be further partitioned, although a complete dominance cannot be obtained.

The interplay between ranking studies and classification is emphasized. Hence, researchers being interested in ranking may additionally be interested in classification. The present study suggests that partial ordering may be helpful also in this second aspect. This can be thought of as being comfortable for any user. However, still more experience and work is needed to further elucidate how, e.g., NN/SVM and partial order methodology can supplement each other. Thus, in order further to elucidate the use of partial ordering for classification studies it appears appropriate to investigating the stability of the dominance and separability matrices with respect to data uncertainty.

The application of dominance and separability matrices implies a further critical point: The central point in many classification algorithms is the metric information inherent in the data matrix. By adoption of the concept of dominance and separability matrices, however, metric information is lost.

A possible way to some extent to recover the metric information, will be to consider not just the poset of observed flowers, but to embed this into a larger poset built upon the set of all profiles (combination of values) obtained by discretizing the input variables in a sufficiently fine way (see for example papers by Fattore and Maggino, [26] Fattore, [19]). Based on this

construction, the mutual ranking probability matrix of the observed profiles, might lead to better separation of classes. However, enlarging the poset unequivocally leads to more difficult computations. Thus, instead of the dominance- and separability matrices a matrix of mutual ranking probabilities could be applied to decide whether or not a linear sum or a complete disjoint union could be stated. However, this procedure is on the one side computationally difficult, and, on the other side, especially the poset of all profiles is losing their structural information, which is evident, if an object oriented poset is applied, as in the case studied here. Therefore, we postpone this kind of analysis for a further publication.

## Acknowledgment

The data applied from this study originate from the UCI Machine Learning Repository [27].

## References

- [1] Iris (1988) Iris data set, <https://archive.ics.uci.edu/ml/datasets/iris> (accessed Mar. 2019)
- [2] Wikipedia. Iris flower data set, [https://en.wikipedia.org/wiki/Iris\\_flower\\_data\\_set](https://en.wikipedia.org/wiki/Iris_flower_data_set), 2019. (accessed Mar. 2020)
- [3] Lars Carlsen and Rainer Bruggemann. Assessing and Grouping Chemicals Applying Partial Ordering Alkyl Anilines as an Illustrative Example, *Comb. Chem. High Throughput Screen.* 21, 349-357, 2018. <https://doi.org/10.2174/1386207321666180604103942>
- [4] Richard P. Stanley. *Enumerative Combinatorics Volume I.* <http://www-math.mit.edu/~rstan/ec/ec1.pdf>, 2011. (accessed Mar. 2020)
- [5] Garrett Birkhoff. *Lattice theory (vol. XXV).* Providence, RI: American Mathematical Society Colloquium Publications Volume XXV, 1984 <https://www.amazon.com/Lattice-American-Mathematical-Colloquium-Publications/dp/B001BTZBYY>(accessed Mar. 2020)
- [6] Helmut Hasse. *Vorlesungen über Klassenkörpertheorie.* Physica-Verlag, Würzburg, 1967.
- [7] Steen Arne Andersson . The lattice Structure of Orthogonal Linear Models and Orthogonal Variance *JSTOR.* 17, 287-319, 1990. [www.jstor.org/stable/4616179](http://www.jstor.org/stable/4616179). (accessed Mar.2020)
- [8] Rodney J. Baxter. Partition Function of the Eight-Vertex Lattice Model. *Annals of Physics* 70, 193-228, 1972. [https://doi.org/10.1016/0003-4916\(72\)90335-1](https://doi.org/10.1016/0003-4916(72)90335-1)
- [9] Oleg Burdakov, Anders Grimvall and Mohammed Hussian. A Generalised PAV algorithm for monotonic regression in several variables, 2014 [https://www.researchgate.net/publication/237808826\\_A\\_GENERALISED\\_PAV\\_ALGORITHM\\_FOR\\_MONOTONIC\\_REGRESSION\\_IN\\_SEVERAL\\_VARIABLES/citation/download](https://www.researchgate.net/publication/237808826_A_GENERALISED_PAV_ALGORITHM_FOR_MONOTONIC_REGRESSION_IN_SEVERAL_VARIABLES/citation/download) (accessed Mar. 2020)

- [10] Deepak Dhar. Entropy and phase transitions in partially ordered sets. *J. Math. Phys.* 19, 1711-1713, 1978. <https://doi.org/10.1063/1.523869>
- [11] Marcello G. Reggiani and Franco E. Marchetti. On Assessing Model Adequacy. *IEEE Transactions on Systems, Man, and Cybernetics SMC-5*, 322-330, 1975. <https://doi.org/10.1109/TSMC.1975.5408407>
- [12] Ernst Ruch. Der Richtungsabstand. *Acta Applicandae Mathematicae* 30, 67-93, 1993. <https://doi.org/10.1007/BF00993343>
- [13] Milan Randić. Design of Molecules with Desired Properties A Molecular Similarity Approach to Property Optimization, 1990, Pages 77-145 in M. A. Johnson and G. M. Maggiora, eds. *Concepts and Applications of Molecular Similarity*. John Wiley & Sons, Inc., New York. <https://www.wiley.com/en-us/Concepts+and+Applications+of+Molecular+Similarity-p-9780471621751> (accessed Mar. 2020)
- [14] Douglas J. Klein. Prolegomenon on Partial Orderings in Chemistry. *MATCH Commun. Math. Comput. Chem.* 42, 7-21, 2000. [http://match.pmf.kg.ac.rs/electronic\\_versions/Match42/match42\\_7-21.pdf](http://match.pmf.kg.ac.rs/electronic_versions/Match42/match42_7-21.pdf)
- [15] Efraim Halfon and Marcello G. Reggiani. On the ranking of chemicals for environmental hazard, *Environ. Sci. Technol.* 20, 1173-1179, 1986. <https://doi.org/10.1021/es00153a014>
- [16] Jean-Pierre Brans and Philippe Vincke. A Preference Ranking Organisation Method (The PROMETHEE Method for Multiple Criteria Decision - Making). *Management Science* 31, 647-656, 1985. <https://doi.org/10.1287/mnsc.31.6.647>
- [17] Bernard Roy. The outranking approach and the foundations of the ELECTRE methods. Pages 155-183 in C.A. Bana e Costa, ed. *Readings in Multiple Criteria Decision Aid*. Springer-Verlag, Berlin, 1990. <https://www.springer.com/gp/book/9783642759376> (accessed Mar. 2020)
- [18] Rainer Bruggemann and Ganapati P. Patil. *Ranking and Prioritization for Multi-indicator Systems - Introduction to Partial Order Applications*. Springer, New York 2011. <https://www.springer.com/gp/book/9781441984760> (Accessed Mar.. 2020)
- [19] Marco Fattore. Partially Ordered Sets and the Measurement of Multidimensional Ordinal Deprivation. *Soc. Indic. Res.* 128, 835-858, 2016. <https://doi.org/10.1007/s11205-015-1059-6>
- [20] Rainer Bruggemann and Lars Carlsen. An attempt to Understand Noisy Posets. *MATCH Commun. Math. Comput. Chem.* 75, 485-510, 2016. [http://match.pmf.kg.ac.rs/electronic\\_versions/Match75/n3/match75n3\\_485-510.pdf](http://match.pmf.kg.ac.rs/electronic_versions/Match75/n3/match75n3_485-510.pdf) (accessed Mar. 2020)
- [21] Lars Carlsen and Rainer Bruggemann. Stakeholders' opinions. Food sustainability as an exemplary case, *Soc. Indic. Res.* 2020 (in press)
- [22] Guillermo Restrepo and Rainer Bruggemann. Dominance and Separability in posets, their application to isoelectronic species with equal total charge. *J. Math. Chem.* 44, 577-602 2008. <https://doi.org/10.1007/s10910-007-9331-x>
- [23] Brian A. Davey and Hillery A. Priestley. *Introduction to Lattices and Order*. 2nd ed., Cambridge University Press, Cambridge, 2002. <https://www.cambridge.org/core/books/introduction-to-lattices-and-order/946458CB6638AF86D85BA00F5787F4F4> (Accessed Mar. 2020)
- [24] Rainer Bruggemann, Lars Carlsen, Kristina Voigt and Ralf Wieland. PyHasse Software for Partial Order Analysis: Scientific Background and Description of Selected Modules. in R. Bruggemann, L. Carlsen, and J. Wittmann (Eds.) *Multi-indicator Systems and Modelling in Partial Order*. Springer, New York, pp 389-423, 2014. <https://www.springer.com/la/book/9781461482222> (Accessed Febr. 2020)
- [25] Peter Koppatz and Rainer Bruggemann. PyHasse and Cloud Computing. in M. Fattore & R. Bruggemann (Eds.). *Partial Order Concepts in Applied Sciences*, pp 291-300. Springer, Cham., 2017. <https://www.springer.com/la/book/9783319454191> (Accessed Mar. 2020)
- [26] Marco Fattore and Filomena Maggino. Partial Orders in Socio-economics. A practical challenge for poset theorists or a cultural challenge for social scientists? Pages 197-214 in R. Bruggemann, L. Carlsen, and J. Wittmann, eds. *Multi-indicator Systems and Modelling in Partial Order*. Springer, New York, 2014. <https://www.springer.com/gp/book/9781461482222> (accessed Dec. 2019).
- [27] Dheeru Dua and Casey Graff (2017), UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science. (accessed Mar. 2020)

# Evaluating Websites of Specialized Cultural Content Using Fuzzy Multi-Criteria Decision Making Theories

Katerina Kabassi, Athanasios Botonis and Christos Karydis

Department of Environment, Ionian University, Minotou Giannopoulou 26, 29100 Zakynthos, Greece

E-mail: kkabassi@ionio.gr, nasbotonis@gmail.com and c.karydis@ionio.gr

**Keywords:** website evaluation, cultural informatics, multi-criteria decision making

**Received:** February 19, 2019

*The museums' conservation labs and the treatments on the artifacts many times are overlooked and are not obvious for the public. Nevertheless, their content, which is more specialized than the content of the main museum, may be of interest to students, researchers, archaeologists, tourists, artists for further education and preservation guidelines purposes. In this paper, we evaluate the electronic presence of museums' conservation labs using both empirical and inspection methods of evaluation. For this purpose, a combination of Analytic Hierarchy Process (AHP) and Fuzzy Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) is used to implement an evaluation experiment that combines inspection and empirical methods of evaluation. The proposed scheme of evaluation that implements a combination of methods and decision making theories for the evaluation of websites with specialized cultural content has been used for evaluating the 29 websites of museum's conservation labs and ranks them taking into account their content, usability, and functionality.*

*Povzetek: Prispevek z metodami umetne intelligence ocenjuje spletne strain muzejev.*

## 1 Introduction

Museums' main role is connected with the exhibition of their artifacts. For this reason, museum websites are mainly concerned with this museum function. Another major work that is being done in a museum environment and is often overlooked by the public is the work carried out within a museum's conservation lab. Consequently, the electronic presence of museum conservation labs is also neglected as a result of ignoring or diminishing the public's consciousness of the important work of preserving collections. Despite this fact, there are museums that have invested in the electronic presentation of their conservation labs. The website of a museum's conservation lab differs from the main website of the museum as it contains more specialized information about the artifacts, the equipment used and the research conducted in the labs.

The existence of a website does not guarantee success. Sometimes the websites are poorly developed. As a result, the interaction is made difficult and the museums may lose attention instead of gaining. Indeed, Dyson and Moran (2000) discussed the importance of creating accessible and usable information resources for online museum projects. Therefore, many researchers have highlighted the need for evaluating websites with cultural content (Cunliffe et al., 2001, van Welie & Klaasse 2004). As a result, most of the evaluations of websites of cultural content are about e-museum websites' evaluations.

There is a plethora of methods and theories that could be used in order to evaluate a museum website (Kabassi 2017), however, not many solutions have been proposed for evaluating websites of specialized cultural

content. A rather common categorization of the proposed evaluation methods is made taking into account the participants of the experiment (Kabassi 2017). Indeed, Lewis & Rieman (1994) as well as Davoli et al. (2005), distinguish methods to empirical methods and inspection methods. Inspection methods are used in experiments that the participants are experts. Empirical methods, on the other hand, are implemented with the participation of different categories of potential users of a museum's website (Kabassi 2017). Each method has different advantages and disadvantages. For example, expert-based evaluations are easier and cheaper compared to empirical ones (Reeves 1993; Karoulis et al. 2006). Empirical methods, on the other hand, may be more successful in capturing end user's perceptions as real users participate in the experiment (Kabassi 2017). However, in this case, the experiment needs a large group of evaluators. This is more complicated and expensive compared to inspection methods but their results are undeniable.

In view of these advantages and disadvantages, some evaluation experiments use both users and experts (Garzotto et al. 1998, Harms & Schweibenz 2001, Vavoula et al. 2009, Sylaiou et al. 2014). In this paper, we have used a combination of inspection and empirical method to evaluate the websites of specialized cultural content, such as the websites of the museums' conservation labs. More specifically, we have used experts to evaluate the importance of the criteria used in the evaluation experiment and estimate their weights and real users for evaluating the different alternative websites. The inspection and the empirical methods are

combined with multi-criteria decision making theories for processing the input and making the essential estimation. The Multi-Criteria Decision Making (MCDM) theories that are used are AHP (Analytic Hierarchy Process) (Saaty 1980) and Fuzzy TOPSIS (Fuzzy Technique for Order of Preference by Similarity to Ideal Solution) (Chen 2000).

AHP aims to analyze a qualitative problem through a quantitative method (Saaty 1980). TOPSIS, on the other hand, aims at ordering evaluation items, which in our case are museum websites, through detecting distance between evaluated objects and optimal solutions (Hwang & Yoon 1981). In the particular evaluation experiment, Fuzzy TOPSIS is used instead of TOPSIS because the theory is used in combination with an empirical method, where real users, and not just experts, participated in the experiment. The empirical method involved users answering a questionnaire with linguistic terms, which is easier for users to comprehend and use. Therefore, Fuzzy TOPSIS (Chen 2000) was used to convert linguistic terms to fuzzy numbers, process the data, making estimations and rank the alternatives.

Taking into account the above, AHP is used to implement the inspection method and fuzzy TOPSIS is used for the implementation of the empirical method. These two theories have different reasoning but seem rather complementary. This combination is mainly reported in the evaluation of websites in e-commerce and, more specifically, in the evaluation of websites of travel agencies (Soleymaninejad et al. 2016) or group-buying (Zhang 2015). Furthermore, Fuzzy AHP has been combined with Fuzzy TOPSIS for evaluating university websites (Nagpal et al. 2015) and e-government sites (Büyüközkan & Ruan 2007). This combination has never been used before in the cultural domain.

## 2 Research aim

Taking into consideration the advantages of the different evaluation methods we have implemented a framework describing an experiment for the evaluation of websites of specialized cultural content combining inspection and empirical methods. For the implementation of the different evaluation methods different multi-criteria decision-making theories have been used. More specifically, we use a combination of different MCDM theories to implement an evaluation experiment that combines inspection and empirical methods of evaluation in order to check the electronic presence of museums' conservation labs.

AHP is combined with an inspection method and Fuzzy TOPSIS with an empirical method of evaluation. This combination is proposed due to the advantages that each method provides. AHP provides the tools to analyse a qualitative problem. The method's ability in making decisions by making pairwise comparison of uncertain, qualitative and quantitative factors and also its ability to model expert opinion (Mulubrhan et al. 2014) are the main reasons for its combination with an inspection method of evaluation. In the particular evaluation

experiment, AHP is used for forming the set of criteria for the evaluation as well as their weights of importance.

Fuzzy TOPSIS, on the other hand, provides adequate tools to analyze the linguistic responses of users in a questionnaire to order the evaluated objects. Indeed, the empirical method that is combined with Fuzzy TOPSIS, involved users answering a questionnaire with linguistic terms, which is easier for users to comprehend and use. For this reason, Fuzzy TOPSIS was considered very suitable for converting linguistic terms to fuzzy numbers, processing the data, making estimations and ranking the alternatives. According to the theory, if the evaluated object is near the optimal solution and far away from the poor solution, it is the best.

Most evaluation experiments of websites in the cultural domain refer to the evaluation of museums' websites not websites of specialized cultural content. The proposed framework that is described in detail in this paper, could be easily applied for the evaluation of other websites of specialized cultural content.

## 3 Multi-criteria decision making methods

MCDM has evolved rapidly over the last decades (Zopounidis 2009). MCDM theories are devoted to the development and implementation of decision support tools and methodologies to confront complex decision problems involving multiple criteria, goals or objectives of conflicting nature (Zopounidis 2000). Various MCDA methods are available, such as AHP, Fuzzy AHP, TOPSIS, Fuzzy TOPSIS, Data Envelopment Analysis (DEA), Multi-attribute utility theory and many more. All these decision methodology approaches differentiate in the way the objectives and alternative weights are determined (Mohamadali & Garibaldi 2011).

Analytic Hierarchy Process (Saaty 1980) is one of the most popular MCDM theories. The choice of AHP amongst other MCDM theories is because it presents a formal way of quantifying the qualitative criteria of the alternatives and in this way removing the subjectivity of the result (Tiwari 2006). Furthermore, the method's ability in making decisions by making a pairwise comparison of uncertain, qualitative and quantitative factors and also its ability to model expert opinion (Mulubrhan et al. 2014) is another important reason for its selection against other alternatives. This method uses the nine-point scale developed by Saaty for evaluation of the goal with the criterion as well as the criterion with the alternative (Mulubrhan et al. 2014).

AHP can be used to implement all the stages of a decision-making process until having the alternatives shorted. However, the main problem of AHP is that the complexity rises with the increase of alternatives; therefore, it is better used when the number of alternatives is limited. A method to resolve this problem is by combining AHP with another theory that manages to process and sort several alternatives without increasing the complexity disproportionately, such as is TOPSIS. This theory calculates the relative Euclidean distance of the alternative from a fictitious ideal alternative. The

alternative closest to that ideal alternative and furthest from the negative-ideal alternative is chosen as the best.

However, the main problem with the use of TOPSIS is that since the evaluation of the alternatives was part of an empirical method, where real users, and not just experts, participated in the experiment, it is difficult for them to evaluate the websites using numbers. Indeed, in many cases, crisp data are inadequate to model real-life situations. The evaluation experiment is using a questionnaire with the linguistic term then Fuzzy TOPSIS (Chen 2000) should be used to process the data, making estimations and rank the alternatives. In this case, fuzzy numbers are used to access the ratings of each alternative with respect to each criterion and Fuzzy TOPSIS is implemented.

## 4 Inspection method for the implementation of AHP

In the first part of the evaluation experiment, an inspection method is implemented using AHP. The steps of the implementation of AHP in an inspection evaluation are the following:

### 1. Developing goal hierarchy

- a. **Forming the overall goal:** The overall goal is to evaluate the museum's conservation labs websites
- b. **Forming the set of criteria:** The criteria for evaluating the websites of the museum's conservation labs have been selected after a review on inspection evaluation experiments of museum websites proposed by Kabassi (2017) and selecting those that seem more appropriate for the particular evaluation.

#### i. **Category 1: Content.** *In this category, all criteria are related to the content of a website.*

1. **c11: Currency/Clarity/Text comprehension.** This criterion checks the currency and the clarity of the text. Currency refers to how successful is the system in providing up-to-date information, and how successfully it can reflect the current state of the world that it represents. Clarity refers to how comprehensible the texts provided to the users are. For this purpose, the quality and the style are checked as well as the way the content is organized and designed in order to make the website credible and trustworthy.
2. **c12: Completeness/Richness.** This criterion checks whether a website has adequate information on the subject.
3. **c13: Quality Content.** This criterion involves the accuracy and understandability of content.
4. **c14: Support of Research.** Checks whether the website provides information for the support of research.

#### ii. **Category 2: Usability.** *All the criteria that are related to Usability.*

1. **c21: Consistency.** Consistency means that similar pieces of information are dealt with in similar fashions (Di Blas et al. 2002).
2. **c22: Accessibility.** Accessibility measures how easily and intuitively accessible is the website's information for any user.
3. **c23: Structure/Navigation.** The structure of the information provided plays an important role in the success of a website. Therefore, the organization of the content pieces should be in such a way that the navigation of the user to the content of the website is easy.
4. **c24: Easy to use/simplicity.** The user interface should be simple and easy to use.
5. **c25: User interface-Overall presentation-Design.** This criterion checks whether the overall presentation is attractive and engaging.
6. **c26: Efficiency.** This criterion shows whether actions within the website can be performed successfully and quickly (Di Blas et al. 2002).

#### iii. **Category 3: Functionality.** *Criteria that are related to the functionality of the website.*

1. **c31: Multilingualism.** the information should be given in more than one language (Di Blas et al. 2002)
2. **c32: Multimedia.** Different media should be used to convey the information (Di Blas et al. 2002)
3. **c33: Interactivity.** This criterion checks whether the content of the website is comprehensive and useful, nicely presented, easy to explore and use.
4. **c34: Adaptivity.** Adaptivity is the ability of the system to adapt to users' characteristics such as needs and interests while adaptability refers to the ability of users to adapt the user interface to their own preferences.

#### c. **Finding the websites to be evaluated:** In this step, the websites of the museums' conservation labs that are going to be evaluated are 29 and are presented in table 1.

#### d. **Forming the hierarchical structure:** In this step, the hierarchical structure is formed so that criteria could be combined in pairs.

2. **Form the set of evaluators:** As an inspection method is used the set of evaluators composites of human experts. Indeed, the correct choice of the expert would give reliable and valid results. Therefore, a double expert (software engineers and domain experts) system is proposed may increase the reliability of the results. As a result the group of evaluators contained 4 professional conservators and 4 software engineers, 3 of which had experience in a University Department of Conservation of Antiquities & Works of Art.

1	Archaeological Museum of Thessaloniki
2	Australian Museum
3	Barberini – Corsini Gallery – Roma
4	Benaki Museum
5	Boston Museum of Fine Arts
6	British Museum
7	Brooklyn museum
8	Byzantine & Christian Museum in Athens
9	De Young museum of Fine Arts
10	Galleria Nazionale d'Arte Moderna
11	Getty Institution
12	Guggenheim Museum
13	Hermitage Museum
14	Metropolitan Museum
15	MoMa
16	Museo Del Prado
17	Museum of Byzantine Culture in Thessaloniki
18	Museum of Islamic Art - Doha
19	National Gallery of Greece
20	National Museum New Delhi
21	NTNU University museum
22	Oriental Institute Museum
23	Rijksmuseum
24	Smithsonian museum
25	Tate Modern
26	Tokyo National Museum
27	University of Michigan Museum of Art
28	Vatican Museum
29	Victoria & Albert Museum

Table 1: The websites of museums' conservation labs that are evaluated.

- Setting up a pairwise comparison matrix of criteria:** In this step, a comparison matrix is formed so that the criteria of the same level are pair-wise compared. More specifically, three matrices are formed. The first compares content, usability and functionality, which are dimensions at the same level and then another one is formed for the sub-criteria of each one of the three dimensions. For example, the matrix of combining the three dimensions is presented in table 2. In the comparison process, a V from the scale that is presented in Table 2 is assigned to the comparison result of two criteria

	Content	Usability	Functionality
Content	1	V	X
Usability	1/V	1	Y
Functionality	1/X	1/Y	1

Table 2: Matrix for the pairwise comparison of the three dimensions.

	Content	Usability	Functionality
Content	1.00	0.46	1.99
Usability	2.16	1.00	2.59
Functionality	0.50	0.39	1.00

Table 3: Matrix for the pairwise comparison of the three criteria of the first level.

‘Content’ and ‘Usability’, then the value of comparison of ‘Usability’ and ‘Content’ is a reciprocal value of V, i.e. 1/V. The value of the comparison of ‘Content’ and ‘Content’ is 1.

Each professional expert combines all four (4) matrices and the final values of each matrix are calculated taking into account the geometric mean of the 8 corresponding values of each matrix’s cell. As a result, the final matrices are built. From the pairwise comparison matrix of the dimensions (Table 3) one can easily derive the fact that usability and content are considered more important than functionality. Tables 4, 5 and 6 present the pairwise comparison matrices of the sub-criteria of content, usability and functionality, respectively. The information collected for the creation of the pairwise comparison matrix of the sub-criteria of usability (Table 5) revealed that museum curators thought that the criteria ‘Content Quality’ and ‘Currency/Clarity/Text comprehension’ were very important whereas experts in usability thought that ‘Overall presentation/Design’ and ‘Structure/Navigation/Orientation’ were more crucial. Finally, in functionality, the opinions of the software engineer, the web designer and the museum curators were in agreement and the pairwise comparison matrix of the sub-criteria of functionality is presented in Table 6.

**Calculating weights of criteria:** After making pairwise comparisons, estimations are made that result in the final set of weights of the criteria. In this step, the principal eigenvalue and the corresponding normalized right eigenvector of the comparison matrix give the relative importance of the various criteria being compared. The elements of the normalized eigenvector are the weights of criteria or sub-criteria. There are several methods for calculating the eigenvector. Multiplying together the entries in each row of the matrix and then taking the nth root of that product approximates the correct answer. The nth roots are summed and that sum is used to normalize the eigenvector elements to add to 1.00. In terms of simplicity, we have used the ‘Priority Estimation Tool’ (PriEst) (Sirah et al. 2015), an open-source decision-making software that implements the Analytic Hierarchy

Process (AHP) method, for making the calculations of AHP (fig. 2). The weights of the criteria are:

$$w_{c1} = 0.292, w_{c2} = 0.534, w_{c3} = 0.174, w_{c23} = 0.214, w_{c24} = 0.213, w_{c25} = 0.164,$$

$$w_{c11} = 0.34, w_{c12} = 0.186, w_{c13} = 0.325, w_{c26} = 0.088, w_{c31} = 0.242, w_{c32} = 0.315,$$

$$w_{c14} = 0.149, w_{c21} = 0.172, w_{c22} = 0.15, w_{c33} = 0.196, w_{c3} = 0.247.$$

	c11: Currency/Clarity/Text comprehension	c12: Completeness/Richness	c13: Quality Content	c14: Support of Research
c11: Currency/Clarity/Text comprehension	1.00	2.24	0.89	2.20
c12: Completeness/Richness	0.45	1.00	0.60	1.47
c13: Quality Content	1.13	1.67	1.00	1.92
c14: Support of Research	0.46	0.68	0.52	1.00

Table 4: Matrix for the pairwise comparison of the sub criteria of Content.

	c21: Consistency	c22: Accessibility	c23: Structure/Navigation	c24: Easy to use/simplicity	c25: User interface-Overall presentation-Design	c26: Efficiency
c21: Consistency	1.00	1.10	0.76	0.83	0.93	2.37
c22: Accessibility	0.91	1.00	0.73	0.68	0.98	1.51
c23: Structure/Navigation	1.31	1.37	1.00	1.13	1.36	2.06
c24: Easy to use/simplicity	1.21	1.47	0.88	1.00	1.62	2.22
c25: User interface-Overall presentation-Design	1.08	1.02	0.74	0.62	1.00	2.26
c26: Efficiency	0.42	0.66	0.48	0.45	0.44	1.00

Table 5: Matrix for the pairwise comparison of the sub criteria of Usability.

	c31: Multilingualism	c32: Multimedia	c33: Interactivity	c34: Adaptivity
c31: Multilingualism	1.00	0.85	1.10	1.00
c32: Multimedia	1.18	1.00	1.70	1.33
c33: Interactivity	0.91	0.59	1.00	0.75
c34: Adaptivity	1.00	0.75	1.34	1.00

Table 6: Matrix for the pairwise comparison of the sub criteria of Functionality.

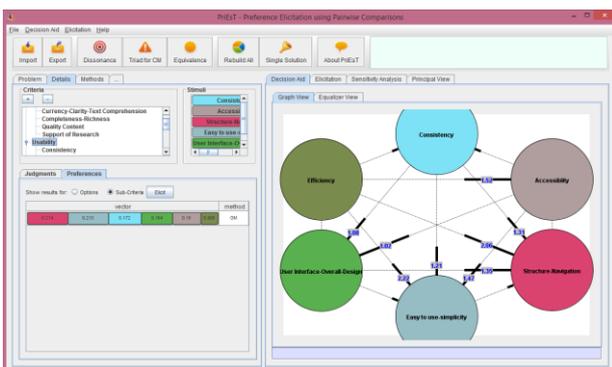


Figure 1: The interface of PriEst.

## 5 Empirical method with the implementation of fuzzy TOPSIS

In the second phase of the evaluation experiment, an empirical method is implemented. For this purpose, a new set of evaluators is formed to contain not only expert users but other categories of users, as well.

1. **Forming a new set of evaluators:** In this phase of the evaluation experiment, the set of evaluators was formed, following the taxonomy of types of users of cultural websites proposed by Sweetnam et al. (2012). More specifically, the final group of

evaluators, which involved professional researchers in conservation, students at advanced undergraduate and postgraduate level, informed users (researchers who are not professional academics but have knowledge of the subject) and the general public.

2. **Assigning values to the criteria:** In order to make this process easier for the user, especially for those that do not have experience in multi-criteria analysis, a questionnaire has been formed. The questionnaire involves a section of demographic questions and then another 29 section, one for each website that was evaluated. Each section contained 14 questions, one for each of the sub-criteria presented in the previous section. The questions provided only multiple-choice answers using the linguistic terms of table 7. The questionnaire was provided electronically using GoogleDocs (Figure 2).
3. **Linguistic terms are transformed to fuzzy numbers.** Each linguistic term is assigned to a fuzzy number, which is a vector like  $\tilde{a} = (a_1, a_2, a_3)$ . The matches are presented in table 7 (Chen 2000).

4. **Construction of the MCDM matrix.** A fuzzy multi-criteria group decision-making problem can be expressed in matrix format. Each element of the matrix is a fuzzy number. However, in order to aggregate all the values of the decision-makers in one single value the geometric mean is used. The geometric mean of two fuzzy numbers  $\tilde{a} = (a_1, a_2, a_3)$  και  $\tilde{b} = (b_1, b_2, b_3)$  is calculated as follows:

$$\tilde{c} = (\sqrt{a_1 b_1}, \sqrt{a_2 b_2}, \sqrt{a_3 b_3}).$$

$$\tilde{D} = \begin{matrix} & C_1 & C & C_n \\ \begin{matrix} A_1 \\ A_2 \\ \vdots \\ A_m \end{matrix} & \begin{pmatrix} \tilde{x}_{11} & \tilde{x}_{12} & \tilde{x}_{1n} \\ \tilde{x}_{21} & \tilde{x}_{22} & \tilde{x}_{2n} \\ \vdots & \vdots & \vdots \\ \tilde{x}_{m1} & \tilde{x}_{m2} & \tilde{x}_{mn} \end{pmatrix} \end{matrix}$$

$$i = 1, 2, \dots, m; j = 1, 2, 3, n,$$

$$\tilde{x}_{i,j} = (a_{i,j}, b_{i,j}, c_{i,j})$$

where i shows the alternative and j shows the criterion. Each  $\tilde{x}_{ij} = (a_{ij}, b_{ij}, c_{ij})$  is a triangular fuzzy number.

5. **Normalisation of fuzzy numbers.** To avoid the complicated normalization formula used in classical TOPSIS, Chen (2000) proposes a linear scale transformation in order to transform the various criteria scales into a comparable scale. The particular normalization method aims at preserving the property that the ranges of normalized triangular fuzzy numbers belong to

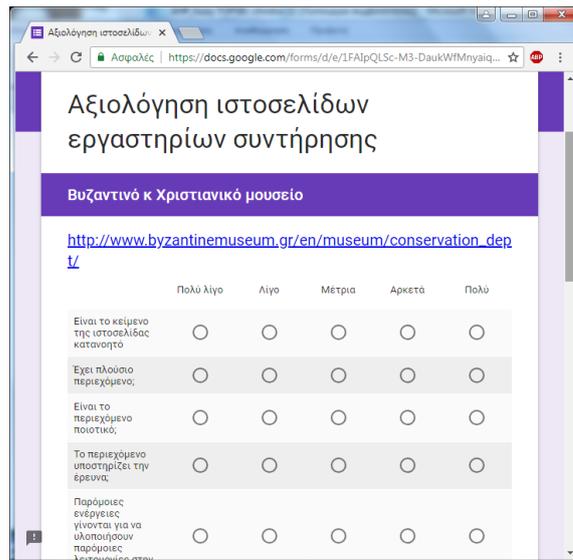


Figure 2: The questionnaire (in greek) of the empirical method.

[0,1]. The normalization of a fuzzy number  $\tilde{x}_{ij} = (a_{ij}, b_{ij}, c_{ij})$  is given by the formula:

$$\tilde{r}_{ij} = \left( \frac{a_{ij}}{c_j^*}, \frac{b_{ij}}{c_j^*}, \frac{c_{ij}}{c_j^*} \right), \text{ where } c_j^* = \max_i c_{i,j}$$

6. **Calculating the weighted normalized fuzzy numbers of the MCDM matrix.** Considering the different importance of each criterion, which is imprinted in the weights of the criteria, the weighted normalized fuzzy numbers are calculated:  $\tilde{u}_{i,j} = \tilde{r}_{i,j}(\bullet)\tilde{w}_j$  and these values are used to construct the weighted normalized fuzzy MCDM matrix  $\tilde{V} = [\tilde{u}_{ij}]_{M \times N}, i = 1, 2, \dots, m; j = 1, 2, 3, \dots, n$

7. **Determination of the Fuzzy Positive-Ideal Solution (FPIS) and the Fuzzy Negative-Ideal Solution (FNIS).** The Fuzzy Positive-Ideal Solution (FPIS) and the Fuzzy Negative-Ideal Solution (FNIS) are calculated as follows:

a. FPIS:  $A^* = \{\tilde{u}_1^*, \tilde{u}_2^*, \dots, \tilde{u}_i^*, \dots, \tilde{u}_n^*\},$

$$\tilde{u}_j^* = (1,1,1)$$

b. FNIS:  $A^- = \{\tilde{u}_1^-, \tilde{u}_2^-, \dots, \tilde{u}_i^-, \dots, \tilde{u}_n^-\},$

$$\tilde{u}_j^- = (0,0,0)$$

Linguistic term	Fuzzy number
Very Poor	(1,1,3)
Poor	(0,1,3)
Fair	(3,5,7)
Good	(7,9,10)
Very Good	(9,10,10)

Table 7: Linguistic terms assigned to fuzzy numbers.

**8. Calculation of the distance of each alternative from FPIS and FNIS**

The distances ( $d_i^*$  και  $d_i^-$ ) of each weighted alternative  $i = 1, 2, \dots, m$  from FPIS and FNIS is calculated as follows:

$$d_i^* = \sum_{j=1}^n d_u(\tilde{u}_{ij}, \tilde{u}_j^*) \quad , \quad i = 1, 2, \dots, m$$

$$d_i^- = \sum_{j=1}^n d_u(\tilde{u}_{ij}, \tilde{u}_j^-) \quad , \quad i = 1, 2, \dots, m$$

where  $d_u(\tilde{a}, \tilde{b})$  is the distance between two fuzzy numbers  $\tilde{a}, \tilde{b}$ . The distance of two fuzzy numbers  $\tilde{a} = (a_1, a_2, a_3)$  and  $\tilde{b} = (b_1, b_2, b_3)$  is calculated

$$d(\tilde{a}, \tilde{b}) = \sqrt{\frac{1}{3}[(\alpha_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2]}$$

**9. Calculation of the closeness coefficient of each alternative.** The closeness coefficient of each alternative  $j$ , is given by the formula

$$CC_i = \frac{d_i^-}{d_i^* + d_i^-}, \quad 0 \leq CC_i \leq 1.$$

According to the values of the closeness coefficient, the ranking order of all the alternatives is determined. The alternative that is closer to FPIS and further from FNIS as  $CC_i$  approaches 1. The values of the closeness coefficient of each alternative and the final ranking of the evaluated websites are presented in table 8.

**6 Discussion**

Museum websites and especially museum conservation labs play an important role in promoting culture. However, a website has to be evaluated so that its effectiveness is verified. Despite its importance, this phase is often omitted by the website life-cycle especially when several criteria are to be checked (Nilashi & Janahmadi 2012). In order to make the evaluation experiment easier for professionals, researchers and students to implement we present in detail the steps that one has to take in order to combine different evaluation methods and different multi-criteria decision-making theories.

The proposed method uses a combination of inspection and empirical method. More specifically, the evaluation experiment is implemented into two phases. In the first part, the inspection method is implemented and in the second part, an empirical method is used. In the first phase, in which the criteria and the weights of the criteria are estimated, expert users can more effectively provide such information. The conclusions are even stronger because both domain and computer experts are used. The

#	Museum Conservation Labs	$CC_i$
1	National Gallery of Greece	0.165445
2	Benaki Museum	0.163536
3	Metropolitan Museum	0.162603
4	Hermitage Museum	0.160753
5	Byzantine & Christian Museum in Athens	0.158755
6	Museo Del Prado	0.150677
7	Vatican Museum	0.150438
8	Archaeological Museum of Thessaloniki	0.148635
9	Victoria & Albert Museum	0.146742
10	Boston Museum of Fine Arts	0.145825
11	Guggenheim Museum	0.144999
12	MoMa	0.144952
13	De Young museum of Fine Arts	0.144765
14	Tokyo National Museum	0.142368
15	Smithsonian museum	0.139851
16	British Museum	0.138384
17	Tate Modern	0.138328
18	Australian Museum	0.134923
19	Rijksmuseum	0.132292
20	Brooklyn museum	0.131841
21	Oriental Institute Museum	0.130455
22	NTNU University museum	0.128826
23	Getty Institution	0.127965
24	University of Michigan Museum of Art	0.126843
25	Museum of Byzantine Culture in Thessaloniki	0.122426
26	Museum of Islamic Art - Doha	0.112785
27	Barberini – Corsini Gallery – Roma	0.105871
28	National Museum New Delhi	0.102325
29	Galleria Nazionale d'Arte Moderna	0.094584

Table 8. The final ranking of the websites based on the values of closeness coefficient of all alternatives.

implementation of the experiment using an inspection method is easier and cheaper than empirical. Despite the advantages of inspection methods, these methods are not appropriate for all kinds of evaluation experiments. For example, in the second part of the experiment, the perception of real users is needed. Therefore, for the second part of the experiment larger group of potential users of the websites was used. This method was more complicated and expensive compared to the previous method but it was considered essential due to the conclusions that had to be extracted.

The inspection method was implemented using AHP. AHP has the ability to model expert opinion

and, therefore, was considered ideal for being combined with an inspection method of evaluation. As a result, AHP was used for the calculation of the weights of the criteria. But AHP is a time-consuming technique because of the mathematical calculations and number of pairwise comparisons which increases as the number of alternatives and criteria increases or changes (Jadhav & Sonar 2011). Since complexity rises with the increase in websites, the number of alternatives that can be compared is limited. This is one of the main reasons for selecting to combine AHP with another theory.

The theory that was selected to implement the empirical method in the second phase of the evaluation experiment was Fuzzy TOPSIS. The complexity of TOPSIS's application does not increase with the same rate of AHP when the number of alternative websites increases. Therefore, the suitability of TOPSIS for the second phase of the website evaluation is inevitable. A main drawback of TOPSIS is that it does not provide a specified way for calculating the weights of criteria as AHP does. Taking into account the advantages and disadvantages of AHP and TOPSIS, these two theories have different reasoning but seem rather complementary. Furthermore, in the case of an empirical method, in which several evaluators are involved that do not have experience in implementing, Fuzzy TOPSIS seems more appropriate. The linguistic terms used in Fuzzy TOPSIS are easier for users to comprehend and use.

The results of the first part of the evaluation revealed that the most important criterion of the first level is Usability, followed by Content. Within the sub-criteria of Content, the Quality of Content was considered the most important criterion. Regarding Usability, the sub-criteria Structure/Navigation and Easy to use/Simplicity are considered almost equally important. As concerned Functionality, the existence of Multimedia is considered the most important criterion.

The results of the second phase of the evaluation revealed that the best website was considered to be the website of the conservation lab of the National Gallery of Greece. The particular website provided rich content related to the activities of the department, the different departments, the equipment, and the staff. Its content is enriched with multimedia. The user interface is well designed and generally, the website is well structured and usable. The website of the Benaki Museum in Athens was also rated high. However, one may be concerned with the fact that two Greek websites were rated first. Although the language is a factor that may have influenced the evaluators, one can also observe that other Greek sites have been ranked in the last five.

Two of the last ranked websites of museums' conservation labs are the websites of the National Museum of New Delhi and the Galleria Nazionale d'Arte Moderna. Their content was poor and there was no information about the staff, the facilities and the equipment. Furthermore, the websites had only a few

photos and no other multimedia. Finally, both websites did not appear to be updated until nowadays.

## 7 Conclusions

Websites of cultural content are targeted to a variety of users (Wubs & Huysmans 2006, Purday 2009, Sweetnam et al. 2012). Therefore, these websites have to address the needs and interests of a variety of users. In order to confirm that a website meets its goals, an evaluation experiment should be implemented. The evaluations are usually complicated procedures that focus on the examination of several different criteria. The particular paper focuses on the evaluation of the websites of museums' conservation labs. The conservation labs in the Museums serve a unique and separate scope and goal inside each institution (i.e. different staff, particular equipment, etc.). Therefore, these websites may differ from the main websites of the museum in terms of content and structure. The framework presented in this paper aims at the evaluation of websites of specialized cultural content in general. The websites of museums' conservation labs that contain specialized cultural content have been used as a testbed to test its functionality.

The main contribution of the particular paper is that it presents a framework for the evaluation of websites of specialized cultural content. This framework combines different methods and different multi-criteria decision-making theories in order to evaluate websites of specialized cultural content. More specifically, it is shown in detail the combination of inspection and empirical methods for evaluating the websites of specialized cultural contents such as the websites of conservation labs in museums. The combination of these two methods is made so as to benefit from the advantages of each method and restrict the disadvantages of each method.

Furthermore, the proposed approach shows how a multi-criteria decision-making theory, namely AHP, is combined with a fuzzy multi-criteria decision-making theory, namely Fuzzy TOPSIS, to evaluate websites of cultural content. AHP's main advantage is that it uses pairwise comparisons of criteria for estimating their weights. However, these pairwise comparisons increase complexity dramatically when the number of alternative websites increases. Therefore, AHP does not seem appropriate for the evaluation of 29 websites. A solution to this problem is given through the use of Fuzzy TOPSIS. The complexity of Fuzzy TOPSIS applications does not increase so dramatically with the increase of alternatives. Furthermore, Fuzzy TOPSIS uses linguistic terms and seems ideal for an experiment where real users, without prior experience in the implementation of multi-criteria decision-making theories, are involved.

It is among our future plans to use this framework in the evaluation of other websites of different specialized cultural content. Furthermore, we aim at trying other MCDM theories and comparing them to find the best combination for the purposes of

evaluation of cultural websites with specialized cultural content.

## References

- [1] Büyüközkan D., D. Ruan. 2007. Evaluating government websites based on a fuzzy multiple criteria decision-making approach. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 15(3): 321–343. <https://doi.org/10.1142/s0218488507004704>
- [2] Chen C.T. 2000. Extensions of the TOPSIS for group decision-making under fuzzy environment. *Fuzzy Sets and Systems*, 114: 1-9. [https://doi.org/10.1016/s0165-0114\(97\)00377-1](https://doi.org/10.1016/s0165-0114(97)00377-1)
- [3] Cunliffe D., E. Kritou, D. Tudhope. 2001. Usability evaluation for museum web sites. *Museum Management and Curatorship*, 19(3): 229-252. <https://doi.org/10.1080/09647770100201903>
- [4] Davoli P., F. Mazzoni, E. Corradini. 2005. Quality Assessment of Cultural Web Sites with Fuzzy Operators, *Journal of Computer Information Systems*, 46(1): 44-57.
- [5] Di Blas N., M.P. Guermant, C. Orsini, P. Paolini. 2002. Evaluating the Features of Museum Websites. In: *Museums and the Web 2002: Selected Papers from an International Conference* (6th, Boston, MA), April 17-20.
- [6] Dyson M. & K. Moran. 2000. Informing the design of Web interfaces to museum collections. *Museum Management and Curatorship*, 18: 391–406. <https://doi.org/10.1080/09647770000501804>
- [7] Garzotto F., M. Matera, P. Paolini. 1998. To use or not to use? Evaluating usability of museum web sites. In *Proceedings of Museums and the Web '98*, Toronto, Canada. Retrieved April 2016: [http://www.museumsandtheweb.com/mw98/papers/garzotto/garzotto\\_paper.html](http://www.museumsandtheweb.com/mw98/papers/garzotto/garzotto_paper.html) <https://doi.org/10.1145/948496.948515>
- [8] Harms I. & W. Schweibenz. 2001. Evaluating the usability of a museum Web site. In D. Bearman, & J. Trant (Eds.), *Museums and the Web* (pp. 43–54). Pittsburgh, PA: Archives and Museum Informatics.
- [9] Hwang C.L., K. Yoon. 1981. *Multiple Attribute Decision Making: Methods and Applications*. Springer-Verlag, New York. <http://dx.doi.org/10.1007/978-3-642-48318-9>
- [10] Jadhav S., R. Sonar, 2011. Framework for evaluation and selection of the software packages: A hybrid knowledge based system approach. *Journal of Systems and Software*, 84: 1394–1407. <https://doi.org/10.1016/j.jss.2011.03.034>
- [11] Kabassi K. 2017. Evaluating Websites of Museums: State of the Art. *Journal of Cultural Heritage* (Elsevier), 24: 184-196. <https://doi.org/10.1016/j.culher.2016.10.016>
- [12] Karoulis S., S. Sylaiou, M. White, 2006. Usability Evaluation of a Virtual Museum Interface, *INFORMATICA*, 17(3): 363–380.
- [13] Lewis C.L., J. Rieman. 1994. *Task-centered User Interface Design: A Practical Introduction*, Boulder: University of Colorado.
- [14] Mohamadali N.A., J. Garibaldi. 2011. Comparing user acceptance factors between research software and medical software using AHP and Fuzzy AHP. In: *The 11th Workshop on Computational Intelligence*, 7 - 9 September 2011, Kilburn Building.
- [15] Mulubrhan F., A. Akmar Mokhtar, M. Muhammad. 2014. Comparative Analysis between Fuzzy and Traditional Analytical Hierarchy Process. *MATEC Web of Conferences* 13. <https://doi.org/10.1051/mateconf/20141301006>
- [16] Nagpal R., D. Mehrotra, P. Kumar Bhatia, A. Sharma, 2015. Rank University Websites Using Fuzzy AHP and Fuzzy TOPSIS Approach on Usability. *International Journal of Information Engineering and Electronic Business*, 1: 29-36. <https://doi.org/10.5815/ijieeb.2015.01.04>
- [17] Nilashi M., N. Janahmadi. 2012. Assessing and Prioritizing Affecting Factors in E-Learning Websites Using AHP Method and Fuzzy Approach. *Information and Knowledge Management* 2(1): 46-61
- [18] Purday J. 2009. Think culture: Europeana.eu from concept to construction. *The Electronic Library*, Vol. 33(2): 170-180. Available: <http://dx.doi.org/10.1108/02640470911004039>
- [19] Reeves T.C. 1993. Evaluating technology-based learning, In G.M. Piskurich (Ed.), *The ASTD Handbook of Instructional Technology*. McGraw-Hill, New York. 15: 1–32.
- [20] Saaty T. 1980. *The analytic hierarchy process*. New York, McGraw-Hill.
- [21] Sirah S., L. Mikhailov, J. A. Keane. 2015. PriEsT: an interactive decision support tool to estimate priorities from pair-wise comparison judgments. *International Transactions in Operational Research*, 22(2): 203–382. <https://doi.org/10.1111/itor.12054>
- [22] Soleymaninejad M., M. Shadifar, A. Karimi. Evaluation of Two Major Online Travel Agencies of US Using TOPSIS Method. *Digital Technologies*, 2(1), (2016), 1-8
- [23] Sweetnam M. S., M. Agosti, N. Orio, C. Ponchia, C.M. Steiner, E.-C. Hillemann, M. Siochrú, S. Lawless. 2012. User needs for enhanced engagement with cultural heritage collections. *Proceedings of Second International Conference TPDL*, Paphos, Cyprus, September, 23 – 27: 64 – 75. Available: <https://dx.doi.org/10.1007/978-3-642-33290-6>. [https://doi.org/10.1007/978-3-642-33290-6\\_8](https://doi.org/10.1007/978-3-642-33290-6_8)
- [24] Sylaiou S., V. Killintzis, I. Paliokas, K. Mania, P. Patias. 2014. Usability Evaluation of Virtual Museums' Interfaces Visualization Technologies.

- In R. Shumaker and S. Lackey (Eds.): VAMR 2014, Part II, LNCS 8526, (2014), 124–133, 2014. © Springer International Publishing Switzerland.  
[https://doi.org/10.1007/978-3-319-07464-1\\_12](https://doi.org/10.1007/978-3-319-07464-1_12)
- [25] Tiwari N. 2006. Using the Analytic Hierarchy Process (AHP) to identify Performance Scenarios for Enterprise Application, Computer Measurement Group, Measure It, 4(3).
- [26] Vavoula G., M. Sharples, P. Rudman, J. Meek, P. Lonsdale, P. 2009. Myartspace: Design and evaluation of support for learning with multimedia phones between classrooms and museums. *Computers and Education*, 53 (2): 286-299.  
<https://doi.org/10.1016/j.compedu.2009.02.007>
- [27] van Welie M., B. Klaasse. 2004. Evaluating Museum Websites using Design Patterns. Technical report number: IR-IMSE-001, December 2004, Vrije Universiteit, Amsterdam
- [28] Wubs H., F. Huysmans. 2006. Click to the past. The Netherland Institute for Social Research. [Online]. Available:  
[http://www.scp.nl/english/Publications/Summaries\\_by\\_year/Summaries\\_2006/Click\\_to\\_the\\_past](http://www.scp.nl/english/Publications/Summaries_by_year/Summaries_2006/Click_to_the_past)
- [29] Zhang W. 2015. Group-Buying Websites Evaluation Model Based on AHP-TOPSIS under the Environment of Multi-Attribute Decision-Making. *International Journal of Multimedia and Ubiquitous Engineering*, 10(7): 31-40.  
<https://doi.org/10.14257/ijmue.2015.10.7.04>
- [30] Zopounidis C. 2000. Foreword: Special issue on artificial intelligence and decision support with multiple criteria. *Computers & Operations research*. Vol. 27, 597-599.  
[https://doi.org/10.1016/s0305-0548\(99\)00107-0](https://doi.org/10.1016/s0305-0548(99)00107-0)
- [31] Zopounidis C. 2009. Knowledge-based multi-criteria decision support. *European journal of operational research*, 195: 827-828.  
<https://doi.org/10.1016/j.ejor.2007.11.026>

# Application of Algorithms with Variable Greedy Heuristics for k-Medoids Problems

Lev Kazakovtsev and Ivan Rozhnov

Reshetnev Siberian State University of Science and Technology

prosp. Krasnoyarskiy Rabochiy 31, Krasnoyarsk 660031, Russia

Siberian Federal University, prosp. Svobodny 79, Krasnoyarsk 660041, Russia

E-mail: levk@bk.ru

**Keywords:** clustering algorithms, VNS, k-medoids, greedy heuristic method

**Received:** March 28, 2019

*Progress in location theory methods and clustering algorithms is mainly targeted at improving the performance of the algorithms. The most popular clustering models are based on solving the  $p$ -median and similar location problems ( $k$ -means,  $k$ -medoids). In such problems, the algorithm must find several points called cluster centers, centroids, medoids, depending on the specific problem which minimize some function of distances from known objects to the centers. In the  $k$ -medoids problem, the centers (medoids) of the cluster must coincide with one of the clustered objects. The problem is NP-hard, and the efforts of researchers are focused on the development of compromise heuristic algorithms that provide a fairly quick solution with minimal error. In this paper, we propose new algorithms of the Greedy Heuristic Method which use the idea of the Variable Neighborhood Search (VNS) algorithms for solving the  $k$ -medoids problem (which is also called the discrete  $p$ -median problem). In addition to the known PAM (Partition Around Medoids) algorithm, neighborhoods of a known solution are formed by applying greedy agglomerative heuristic procedures. According to the results of computational experiments, the new search algorithms (Greedy PAM-VNS) give more accurate and stable results (lower average value of the objective function and its standard deviation, smaller spread) in comparison with known algorithms on various data sets.*

*Povzetek: Avtorji predlagajo nove algoritme za reševanje problema lokacije  $k$ -medoidov in gručenja.*

## 1 Introduction

The rapid development of artificial intelligence systems using, inter alia, methods of automatic data grouping (clustering) and methods of location theory, as well as increasing requirements for economic efficiency in all branches, creates a request for the creation of new algorithms with higher requirements for accuracy of the result.

The attempts to discover a universal and, at the same time, exact method for solving most popular location and clustering problems ( $k$ -means,  $k$ -medoids, etc.), which guarantees the global optimum of the objective function, in the case of a large amount of input data has been recognized as unpromising. The efforts of researchers focused on the development of compromise heuristic algorithms that give a quick solution [1]. The heuristic algorithms or procedures, also called "heuristics" in the literature, are algorithms that do not have a rigorous justification, but gives an acceptable solution to the practically important problems. The so-called "greedy" algorithms are also heuristics. On each iteration, the greedy algorithm selects the best solution from a certain neighborhood (subset of intermediate solutions). At the same time, some of the practically important clustering problems require such a solution which is very close to the exact solution of the problem, and also stable during

repeated runs of the randomized algorithm, reproducible and, therefore, verifiable. The problems should be solved online within a limited time. Such problems include, for example, the problem of forming special batches of semiconductor devices in the specialized testing centers [1], where the need to obtain stable results is due to the requirement of reproducibility and verifiability of calculation results that are part of the production process involving two parties with different interests: the manufacturer and the test Centre.

The ensemble (collective) approach [2] allows reducing the dependence of the final decision on the selected parameters of the original models and algorithms and obtaining a more stable solution [3], or isolating "controversial" objects for which different clustering models give a contradictory result, into a separate class. The  $k$ -medoids problem is a convenient model for building clustering algorithm ensembles due to the adaptability of the model to the use of various measures of the distance between objects.

The overall aim of the continuous location problem [4] is to find the location of one or several points (centers, centroids, medoids) in continuous space. There is an intermediate class of problems that are actually discrete (the number of possible locations of the searched

points is finite), operating with concepts characteristic of the continuous problems. In particular, such is the the k-medoids problem [5, 6] (also called the discrete p-median problem [7] in the scientific literature). The main parameters of all such problems are the coordinates of the objects and the distances between them [8-10]. The aim of the continuous p-median problem [8] is to find k points (centers, centroids, medians, cluster medoids), such that the sum of weighted distances from N known points, called demand points, consumers, objects or data vectors depending on the formulation of a specific problem, to the nearest of the k centers reaches its minimum.

The allocation problems with Euclidean, Manhattan (rectangular), Chebyshev metrics are well studied (all these metrics are particular cases of metrics based on Minkowski lp-norms [11]), and many algorithms have been proposed for solving the Weber problem for these metrics. In particular, the well-known Weisfeld procedure [12] was generalized for metrics based on Minkowski norms.

If the distance is Euclidean  $L(X_j, A_i) = \sqrt{\sum_{k=1}^d (x_{j,k} - a_{i,k})^2}$ , we have the p-median problem. Here,  $X_j = (x_{j,1}, \dots, x_{j,k}) \forall j = \overline{1, p}$ ,  $A_i = (a_{i,1}, \dots, a_{i,d}) \forall i = \overline{1, N}$ . If the squared Euclidean metric is used,  $L(X_j, A_i) = \sum_{k=1}^d (x_{j,k} - a_{i,k})^2$ , we have the k-means problem. Vectors  $A_1, \dots, A_N$  are data vectors in a d-dimensional space,  $A_i = (a_{i,1}, \dots, a_{i,d})$ ,  $A_i \in \mathbb{R}^d$ ,  $A_i \in \mathbb{R}^d$ .

In the k-medoids model and problem, cluster centers  $X_j = (x_{j,1}, \dots, x_{j,k})$  called medoids, are searched among the known points  $A_i$ , and this is a discrete optimization problem.

The most popular algorithm for the k-medoid problem, Partitioning Around Medoids (PAM) algorithm, was created by L. Kaufman and P. J. Rousseeuw [13]. It is very similar to the k-means algorithm. Both algorithms divide a lot of objects into groups (clusters) and both are based on attempts to minimize the error (total distance) on each iteration. The PAM algorithm works with medoids, objects that are part of the original set and representing the cluster in which they are included, and the k-means algorithm works with centroids, which are artificially created objects representing a cluster. The PAM algorithm divides a set of N objects into k clusters (k is a parameter of the algorithm). This algorithm operates the pre-calculated distance matrix between objects, its aim is to minimize the distance between the medoid of each cluster and other objects included in the same cluster.

For discrete optimization problems, the local search methods are the most natural and visual [14]. Such problems include the location problems, building networks, schedules, etc. [15-18]. The standard local descent algorithm starts with some initial solution  $x_0$  (in our case, the initial set of medoids) chosen randomly or with the use of some additional algorithm. At each step of a local descent, the current solution is transformed into to the neighboring solution with a smaller value of the

objective function until a local optimum is reached. At each step of local descent, the function of neighborhood  $O$  defines a set of possible directions of local search. Very often this set consists of several elements and there is a certain freedom in choosing the next solution. On the one hand, when choosing a neighborhood, it is desirable to have a set of  $O(X)$  as small as possible in order to reduce the complexity of a single step. On the other hand, a wider neighborhood can lead to a better local optimum. A possible way to resolve this contradiction is to develop complex neighborhoods, the size of which can be varied during local search [19].

In this paper, we propose the use of local search algorithms that contain greedy agglomerative heuristic procedures, as well as the well-known PAM algorithm, using an idea of the Variable Neighborhood Search (VNS) [20]. It is shown that new VNS algorithms have advantages over the standard PAM algorithm and are competitive in comparison with the known genetic algorithms of the Greedy Heuristics Method for the considered problem [21].

## 2 Idea of new algorithms

Local search methods have been further developed into metaheuristics [22]. We consider one of them, called the Variable Neighborhoods Search [23, 24]. The idea is to systematically vary the neighborhood function during a local search. Flexibility and efficiency explain its competitiveness in solving NP-hard problems, in particular, p-median problems [25], clustering and location problems [26, 27].

Let us denote by  $N_k$ ,  $k=1, \dots, k_{max}$ , the finite set of neighborhood functions preselected for local search. The proposed method with variable neighborhoods relies on the fact that a local minimum in one neighborhood is not necessarily a local minimum in another neighborhood, and the global minimum is the local minimum in all neighborhoods [14]. In addition, on average, local minima are closer to the global than a randomly selected point, and they are located close to each other. This allows us to narrow the search area for a global optimum using information about local optimums already detected. This hypothesis forms the basis for various crossover operators for genetic algorithms [28] and other approaches.

The deterministic local descent with variable neighborhoods (VND) implies a fixed order of changing neighborhoods and finding a local minimum relative to each of them. Probabilistic local descent with variable neighborhoods differs from the previous VND method by a random selection of points from the neighborhood  $O_k(X) \in N_k$ . The stage of finding the best point in the neighborhood is omitted. The probabilistic algorithms are most productive in solving problems of large dimension, when the use of a deterministic version requires too much machine time to perform one iteration.

The basic local search scheme with variable neighborhoods is a combination of the two previous options [23].

VNS algorithm

Step 1. Choose the neighborhoods  $O_k$ ,  $k = 1, \dots, k_{max}$ , and the starting point  $x$ .

Step 2. Repeat until the stopping criterion is satisfied.

2.1.  $k \leftarrow 1$ .

2.2. Repeat until  $k \leq k_{max}$ :

2.2.1. Randomly select a point  $x' \in O_k(x)$ ;

2.2.2. Apply a local descent from the starting point  $x'$ . The resulting local optimum is denoted by  $x''$ ;

2.2.3. if  $F(x'') < F(x)$ , then it is assumed that  $x \leftarrow x''$ ,  $k \leftarrow 1$ , otherwise  $k \leftarrow k + 1$ .

We can use the time limitation or maximum number of iterations as the stop criterion. In the case of large-scale problems, the complexity of performing one iteration becomes very large and new approaches are needed to develop effective local search methods.

A popular idea in solving continuous clustering problems is the use of genetic algorithms (GA) and other evolutionary approaches to improve the results of local search [29-31]. Many of these evolutionary algorithms recombine the initial solution obtained by one of the simple local search algorithms.

The PAM procedure consists of two phases: BUILD and SWAP:

- In the BUILD phase, primary clustering is performed, during which  $k$  objects are successively selected as medoids.

- The SWAP phase is an iterative process in which the algorithm makes attempts to improve some of the medoids. At each iteration of the algorithm, a pair is selected (medoid and non-medoid) such that replacing the medoid with a non-medoid object gives the best value of the objective function (the sum of the distances from each object to the nearest medoid). The procedure for changing the set of medoids is repeated as long as there is a possibility of improving the value of the objective function.

Algorithm 1 (PAM procedure).

Initialization (if needed):

1. Select  $k$  objects as a set of medoids (if such set is not given).

2. Build a distance matrix if needed.

*Build Phase:*

3. Assign each object to the nearest medoid.

*Swap Phase:*

4. For each cluster, find objects that reduce the average distance, and if there are such objects, select those that reduce it most strongly, as a medoid.

5. If at least one medoid has changed, return to Step 3, otherwise stop the algorithm.

The following algorithm is the basic algorithm of the Greedy Heuristics Method [1] for clustering problems. His idea is that initially some unacceptable solution with an excessive number of centers / medoids is selected, which is then gradually reduced to a solution with a given number of centers.

Algorithm 2. Basic greedy agglomerative heuristic procedure.

Required: the initial number of clusters  $K$ , the required number of clusters  $k < K$ .

1. Randomly choose an initial solution with  $K$  cluster centers  $S = \{X_1, \dots, X_k\}$  if it is not given.

2. Execute Algorithm 1 with the initial solution  $S$ , store a new (improved) solution to  $S$ .

3. If  $K = k$ , then stop.

4. For each  $i' \in \{1, K\}$  perform:

4.1. Get a truncated set  $S \leftarrow S \setminus \{X_{i'}\}$ .

4.2. Run Algorithm 1 with the initial solution  $S'$ . In this case, Algorithm 1 is limited to only one iteration. Store the achieved value of the objective function (1) to  $F'_{i'}$ .

4.3. Next iteration of loop 4.

5. Find the index  $i'' = \arg \max_{i' \in \{1, k\}} F'_{i'}$ .

6. Get a truncated set  $S \leftarrow S \setminus \{X_{i''}\}$ , improve it with Algorithm 1, then go to step 3.

This procedure forms the basis for three new procedures that use the centers/medoids of some second known feasible solution to compile an intermediate infeasible solution with an excessive number of clusters  $K$ .

Algorithm 3. Greedy procedure # 1.

Given: multiple cluster centers  $S' = \{X'_1, \dots, X'_k\}$  and  $S'' = \{X''_1, \dots, X''_k\}$ .

1. For each  $i' \in \{1, k\}$  perform:

1.1. Combine element by element sets  $S'$  and  $S''$ :  $S \leftarrow S' \cup \{X''_{i'}\}$ .

1.2. Run the basic greedy heuristic (Algorithm 2) with  $S$  as the initial solution. The result obtained (the resulting set, as well as the value of the objective function) is memorized.

2. Return the best (by the value of the objective function) solution among the solutions obtained in step 1.2.

A simpler, but more demanding in terms of computing resources version of the similar algorithm is presented below.

Algorithm 4. The greedy procedure # 2.

1. Form a unified set  $S \leftarrow S' \cup S''$ .

2. Run Algorithm 2 with  $S$  as the initial solution.

An intermediate variant is suggested in [21]. There, the Algorithm 2 starts from the set  $S'$  united with a randomly chosen subset of  $S$ .

Algorithm 5. The greedy procedure # 3.

1. Choose random  $r' \in [0, 1)$ . Assign  $r \leftarrow [(k/2 - 2)r'^2] + 2$ . Here,  $[.]$  is the integer part.

2. Repeat  $k - r$  times:

2.1. Form a randomly selected subset of  $S'''$  of the  $r$  elements of the set  $S''$ . Join the sets  $S \leftarrow S' \cup S'''$ .

2.2. Run Algorithm 2 with this set  $S$  as the initial solution.

3. Return the best (in terms of the objective function) among the solutions obtained in step 2.2.

These heuristic procedures, which are local search algorithms in the neighborhood of a known (“parent”) solution represented by the set  $S'$ , can be used as a part of various global search strategies. At the same time, as the neighborhoods of this solution  $S'$  are formed by adding elements from the other known solution  $S''$  and eliminating the excessive elements from the unified solution using the basic greedy agglomerative heuristic procedure.

The search in such neighborhoods is made by Algorithms 3-5. Thus, these algorithms search in some neighborhoods of the solution  $S'$ , and the second known solution  $S''$  is a randomly selected parameter of this neighborhood. The general idea of the new Variable Neighborhood Search algorithms neighborhoods for solving the k-medoids problem is given below:

Algorithms 6. PAM-VNS.

1. Run Algorithm 1 from a random initial solution, store the resulting set of medoids to  $S$ .
2. Assign  $O \leftarrow O_{start}$  //comment:  $O_{start}$  is the initial number of neighborhood type).
3. Assign  $i \leftarrow 0, j \leftarrow 0$ ; (the number of unsuccessful iterations in a particular neighborhood and as a whole by the algorithm).
4. Run Algorithm 1 from the random initial solution, get the solution  $S'$ .
5. Depending on the value of  $O$  (values 1, 2 or 3 are allowed), run Algorithm 3, 4 or 5 with the initial solutions  $S$  and  $S'$ .
6. If the result (by the objective function value) is better than  $S$ , then replace  $S$  with this new result, assign  $i=0, j=0$ , go to Step 5.
7. Assign  $i \leftarrow i+1$ ;
8. If  $i < i_{max}$ , then go to Step 4.
9. Assign  $i \leftarrow 0, j \leftarrow j + 1$ . Switch to a new neighborhood type:  $O=O+1$ ; if  $O>3$ , then assign  $O=1$ ;
10. If  $j > j_{max}$ , or other stop conditions are satisfied (maximum running time), then STOP. Otherwise, go to Step 5.

The values of the two control parameters are important: the number of ineffectual searches in the current neighborhood  $i_{max}$ , and the number of ineffectual switching of the neighborhoods  $j_{max}$ . We used the values  $i_{max} = 2k, j_{max} = 2$ .

In addition, important control parameter  $O_{start}$ , which specifies the number of the starting neighborhood type. We performed our experiments with all its possible values (1, 2 and 3). Depending on this value, the algorithms are designated below, respectively, PAM-VNS1, PAM-VNS2, PAM-VNS3. In these versions of Algorithm 6, the number of elements in  $S'$  is equal to the number of elements in  $S$ :  $|S|=|S'|=k$ . In special versions called PAM-VNS1-R, PAM-VNS2-R, PAM-VNS3-R,

the number of elements (medoids) in  $S'$  is chosen randomly,  $S' \in \{2, 2k\}$ .

### 3 Computational experiments

In the description of the computation experiments, we used the following abbreviations of the algorithm names: PAM is the classical PAM algorithm in multi-start mode; PAM-VNS1, PAM-VNS2, PAM-VNS3, PAM-VNS1-R, PAM-VNS2-R, and PAM-VNS3-R are variations of Algorithm 6; GA-FULL is the genetic algorithm with a greedy heuristic for the k-medoids problem [1]; GA-ONE is a new genetic algorithm with greedy heuristic [1] where Algorithm 3 is used as a crossing-over procedure.

As test data sets for our experiments, we used the results of non-destructive test tests of prefabricated production batches of semiconductor devices (Tables 1-7), datasets from repositories UCI [32] and Clustering basic benchmark [33]. In our experiments, we used the DEXP computing system (4-core Intel® Core™ i5-7400 CPU 3.00 GHz, 8 GB of RAM).

For all data sets, 30 attempts were made with each of the 9 algorithms. In every attempt, we fixed the best achieved results. The best values of the objective function (minimum value, mean value, median value and standard deviation) are highlighted in bold italics, the smallest of the best values is additionally highlighted. We used the T-test and the Wilcoxon signed rank test [34, 35] (significance level 0.01 for both tests). Note: “↑”, “↑↑”: the advantage of the best of new algorithms over known algorithms is statistically significant (“↑” for the t-test, and “↑↑” for the Wilcoxon test); “↓”, “↓↓”: the disadvantage of the best new algorithms compared to known algorithms is statistically significant; “↕”, “↕↕”: advantage or disadvantage is statistically insignificant.

Table 4 presents the results of the new algorithm in comparison with known evolutionary algorithms that have worked well in solving this problem [1].

In Table 4, we use the following abbreviations [1]:

- GA a genetic algorithm with uniform stochastic crossingover procedure,
- GAGH is the genetic algorithm with greedy heuristic #3 as crossingover procedure,
- LS is the local search by PAM algorithm in

Algorithm	Objective function value (sum of distances)			
	Min (the best attempt)	Average among 30 attempts	Median	Standard deviation
PAM	1 654,4	1 677,4	1679,5	12,2445
PAM-VNS1	<b><i>1 554,3</i></b>	<b><i>1 566,7</i></b>	<b><i>1565,7</i></b>	7,4928
PAM-VNS2	1 558,0	<b><i>1 566,1</i></b>	<b><i>1566,5</i></b>	<b><i>5,0686</i></b>
PAM-VNS3 ↑↑	<b><i>1 555,1</i></b>	<b><i>1 563,9</i></b>	<b><i>1564,9</i></b>	<b><i>3,9161</i></b>
GA-FULL	1 599,2	1 637,6	1636,2	25,5365
GA-ONE	1 589,9	1 614,8	1615,4	13,5342

Table 1: Comparative results of computational experiments with data set 30T122A (767 data vectors, 13 attributes) 10 clusters, 60 seconds for each attempt, 30 attempts, Manhattan distance.

Algorithm	Objective function value (sum of distances)			
	Min (the best attempt)	Average among 30 attempts	Median	Standard deviation
PAM	50 184,0	50 883,7	50 693,0	472,441
PAM-VNS1 ↑↑	<b>45 440,4</b>	<b>45 553,0</b>	<b>45 496,6</b>	<b>95,800</b>
PAM-VNS2	45 453,7	45 657,7	45 648,4	153,329
PAM-VNS3	<b>45 444,4</b>	<b>45 637,9</b>	<b>45 594,4</b>	177,586
GA-FULL	46 660,9	48 391,2	48 341,9	845,084
GA-ONE	47 081,3	48 125,9	47 965,0	766,566

Table 2: Comparative results of computational experiments with data set 5514BC1T2-9A5 (91 data vectors, 173 attributes) 10 clusters, 60 seconds for each attempt, 30 attempts, Manhattan distance.

Algorithm	Objective function value (sum of distances)			
	Min (the best attempt)	Average among 30 attempts	Median	Standard deviation
PAM	50 184,0	50 883,7	50 693,0	472,441
PAM-VNS1 ↑↑	<b>45 440,4</b>	<b>45 553,0</b>	<b>45 496,6</b>	<b>95,800</b>
PAM-VNS2	45 453,7	45 657,7	45 648,4	153,329
PAM-VNS3	<b>45 444,4</b>	<b>45 637,9</b>	<b>45 594,4</b>	177,586
GA-FULL	46 660,9	48 391,2	48 341,9	845,084
GA-ONE	47 081,3	48 125,9	47 965,0	766,566

Table 3: Comparative results of computational experiments with data set 1526TL1 (1234 data vectors, 157 attributes) 10 clusters, 60 seconds for each attempt, 30 attempts, Manhattan distance.

- multistart mode,
- GA FIX is the genetic algorithm with recombination of fixed-length subsets [36],
- Determ.GH is the deterministic algorithm with greedy heuristic [1] built on the principles of the Information Bottleneck Clustering.
- For some of the datasets, we performed our computational experiments with various number of clusters and various distance metrics (Tables 5-7).

For the genetic algorithms, we used the population size NPOP starting from NPOP=20. In [21], authors show that smaller populations (NPOP<10) in the genetic algorithms with the greedy agglomerative crossover procedure decrease the accuracy of the result, and larger populations (NPOP>50) slow down the algorithm which also decreases the accuracy.

In all genetic algorithms, we used the simple tournament selection. Traditionally [30], such algorithms do not contain any mutation operator.

### 4 Conclusion

The results of our computational experiments showed that the new search algorithms in alternating neighborhoods (PAM-VNS) can outperform known algorithms and give more stable results (a lower median and average values and / or standard deviation of the

Algorithm	Objective function value (sum of distances)			
	Min (the best attempt)	Average among 30 attempts	Median	Standard deviation
PAM	64 232,0	66 520,2	66 776,2	991,994
PAM-VNS1 ↓↓	<b>55 361,8</b>	<b>55 363,9</b>	56 004,1	<b>2,457</b>
PAM-VNS2	<b>55 361,8</b>	55 858,4	55 904,3	359,416
PAM-VNS3	55 383,8	55 755,0	<b>55 662,5</b>	353,947
GA-FULL	58 789,3	60 629,5	61 069,2	1187,09
GA-ONE	58 300,2	60 165,4	59 689,1	1388,62
GAGH+LS	<b>55 361,8</b>	55 364,1	55 754,2	6,2204
GAGH	<b>55 361,8</b>	<b>55 361,8</b>	<b>55 622,3</b>	<b>7,8E-12</b>
GA FIX	<b>55 361,8</b>	55 452,7	55 814,1	240,563
GA classical	<b>55 361,8</b>	55 364,1	<b>55 638,9</b>	6,220
Determ GH	55 998,2	55 998,2	56 199,4	<b>0,000</b>

Table 4: Comparative results of computational experiments with data set 1526TL1 (1234 data vectors, 157 attributes) 10 clusters, 60 seconds for each attempt, 30 attempts, squared Euclidean distance.

Algorithm	Objective function value (sum of distances)			
	Min (the best attempt)	Average among 30 attempts	Median	Standard deviation
PAM	2 688,57	2 704,17	2 702,58	12,3308
PAM-VNS1 ↑↑	<b>2 607,21</b>	<b>2 607,25</b>	<b>2 607,21</b>	<b>0,1497</b>
PAM-VNS2	<b>2 607,21</b>	<b>2 607,43</b>	<b>2 607,21</b>	<b>0,4303</b>
PAM-VNS3	<b>2 607,21</b>	<b>2 607,34</b>	<b>2 607,21</b>	<b>0,4159</b>
GA-FULL	<b>2 608,22</b>	2 624,97	2 625,77	9,5896
GA-ONE	<b>2 608,69</b>	2 625,18	2 624,57	10,7757

Table 5: Comparative results of computational experiments with data set Ionosphere (351 data vectors, 35 attributes) 10 clusters, 60 seconds for each attempt, 30 attempts, Manhattan distance.

Algorithm	Objective function value (sum of distances)			
	Min (the best attempt)	Average among 30 attempts	Median	Standard deviation
PAM	319,84	<b>343,44</b>	<b>346,23</b>	<b>15,300</b>
PAM-VNS1	278,63	390,43	367,03	82,609
PAM-VNS2	333,26	471,15	450,34	100,259
PAM-VNS3	<b>273,91</b>	354,98	<b>352,75</b>	54,244
PAM-VNS1-R	301,91	428,14	398,28	129,216
PAM-VNS2-R	384,62	475,92	470,64	53,097
PAM-VNS3-R ↑↑	<b>265,96</b>	<b>325,49</b>	<b>317,94</b>	42,144
GA-FULL	315,57	383,41	365,04	60,149
GA-ONE	343,21	433,01	424,73	66,036

Table 6: Comparative results of computational experiments with data set Mopsi-Joensuu (6015 data vectors, 2 attributes) 20 clusters, 60 seconds for each attempt, 30 attempts, Euclidean distance.

objective function, a smaller spread of the achieved values) and, consequently, better performance in comparison with known algorithms. The comparative efficiency of the new algorithm and its modifications on several data sets has been experimentally proven.

Algorithm	Objective function value (sum of distances)			
	Min (the best attempt)	Average among 30 attempts	Median	Standard deviation
PAM	10 763,0	10 822,4	10 833,0	<b>47,127</b>
PAM-VNS1	10 357,0	<b>10 530,9</b>	<b>10 563,5</b>	122,962
PAM-VNS2	10 803,0	11 107,1	11 079,5	174,118
PAM-VNS3	10 429,0	10 594,6	<b>10 575,0</b>	114,719
PAM-VNS1-R	10 400,0	10 659,0	10 664,5	161,298
PAM-VNS2-R	10 891,0	11 097,0	11 096,5	187,911
PAM-VNS3-R $\updownarrow$	<b>10 310,0</b>	10 623,3	<b>10 597,5</b>	214,129
GA-FULL	<b>10 252,0</b>	<b>10 381,3</b>	<b>10 393,0</b>	<b>72,911</b>
GA-ONE	10 944,0	11 098,0	11 064,5	112,081

Table 7: Comparative results of computational experiments with data set Chess (3196 data vectors, 37 binary attributes) 50 clusters, 60 seconds for each attempt, 30 attempts, squared Euclidean distance.

However, for some of the considered datasets, the genetic algorithms show their advantages over new algorithms. Genetic algorithms show the best results in the case of a complex relief of the objective function, the presence of plateaus, due to diversity in the population, as revealed in some of the cases considered. One of the most important shortcomings of such algorithms is the possible “dumping” of the entire population into the region of attraction of a single local minimum. Probably, new VNS-algorithms allow us to avoid this drawback due to the continuous random generation of new solutions, which are parameters of the search neighborhood, and allow us to “jump out” of the attraction region.

Therefore, the interest for further research is the combined approach, combining the presence of a certain population with the possibility of generating random solutions, which play a role similar to that of the mutation operator in traditional genetic algorithms.

## Acknowledgement

Results were obtained in the framework of the state task No. 2.5527.2017/8.9 of the Ministry of Education and Science of the Russian Federation.

## References

- [1] Kazakovtsev L.A. (2016) *The greedy heuristics method for systems of automatic grouping of objects*. Diss ... Dr. tech. of science. Krasnoyarsk. Siberian Federal University.
- [2] Ghosh J., Acharya A. (2011) Cluster ensembles. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Vol. 1(4), pp.305–315. <https://doi.org/10.1002/widm.32>.
- [3] Rozhnov I., Orlov V., Kazakovtsev L. (2018) Ensembles of clustering algorithms for problem of detection of homogeneous production batches of semiconductor devices. *CEUR-WS*, Vol.2098, pp.338-348. <http://ceur-ws.org/Vol-2098/paper29.pdf>
- [4] Drezner Z., Hamacher H. (2004) *Facility location: applications and theory*. Berlin:Springer-Verlag.
- [5] Struyf A., Hubert M., Rousseeuw P. (1997) Clustering in an Object-Oriented Environment. *Journal of Statistical Software*, Issue 1 (4), pp.1-30. <https://doi.org/10.18637/jss.v001.i04>.
- [6] Kaufman L., Rousseeuw P.J. (1990) Finding groups in data: an introduction to cluster analysis. New York:Wiley. <https://doi.org/10.1002/9780470316801>
- [7] Moreno-Perez J.A., Roda Garcia J.L., Moreno-Vega J.M. (1994) *A Parallel Genetic Algorithm for the Discrete p-Median Problem*. *Studies in Location Analysis*, Issue 7, pp.131-141.
- [8] Wesolowsky, G. (1993) The Weber problem: History and perspectives. *Location Science*, No.1, pp.5-23.
- [9] Drezner Z., Wesolowsky G.O. (1978) A Trajectory Method for the Optimization of the Multifacility Location Problem with lp Distances. *Management Science*, Vol.24, pp.1507-1514. <https://doi.org/10.1287/mnsc.24.14.1507>
- [10] Farahani R., Hekmatfar M. (2009). *Facility location: Concepts, models, algorithms and case studies*. Berlin Heidelberg:Springer-Verlag. <https://doi.org/10.1080/13658816.2010.528422>
- [11] Deza M.M., Deza E. (2013) Metrics on Normed Structures. *Encyclopedia of Distances*. Berlin Heidelberg: Springer, pp.89-99. <https://doi.org/10.1007/978-3-642-30958-85>.
- [12] Weiszfeld, E. (1937) Sur le point sur lequel la somme des distances de n points donnees est minimum. *Tohoku Mathematical Journal*, Vol.43, No.1, pp.335-0386. <https://link.springer.com/article/10.1007/s10479-008-0352-z>
- [13] Kaufman L. and Rousseeuw P.J. (1987) Clustering by means of Medoids. *Statistical Data Analysis Based on the L1-Norm and Related Methods*, Springer US. pp. 405-416.
- [14] Kochetov Yu., Mladenovic N., Hansen P. (2003) Local search with alternating neighborhoods. *Discrete analysis and operations research*, Series 2, Vol.10(1), pp.11-43.
- [15] Nicholson T. A. J. (1965) A sequential method for discrete optimization problems and its application to the assignment, traveling salesman and tree scheduling problems. *J. Inst. Math. Appl*, Vol.13, pp.362-375. <https://doi.org/10.1093/imamat/3.4.362>
- [16] Page E. S. (1965) On Monte Carlo methods in congestion problems. I: Searching for an optimum in discrete situations. *Oper. Res.* Vol.13(2), pp. 291-299. <https://doi.org/10.1287/opre.13.2.291>
- [17] Kernighan B. W., Lin S. (1970) An efficient heuristic procedure for partitioning graphs. *Bell Syst. Tech. J.* Vol.49. pp.291-307. <https://doi.org/10.1002/j.1538-7305.1970.tb01770.x>
- [18] Rastrigin L.A. (1978) Random search - specificity, stages of history and prejudice. *Questions of*

- cybernetics. M.: Nauch. Council on the complex problem "Cybernetics" of the USSR Academy of Sciences, Vol. 33, pp.3-16.
- [19] Kochetov Yu.A. (2010) Local search methods for discrete placement problems. *Dis. Doctors of Physical and Mathematical Sciences*. Novosibirsk.
- [20] Hansen P., Mladenovic N., Bruke E.K., Kendall G. (2014) *Variable Neighborhood Search. Search Methodology*. Springer US. P.211-238. [https://doi.org/10.1007/0-387-28356-0\\_8](https://doi.org/10.1007/0-387-28356-0_8).
- [21] Kazakovtsev, L.A., Antamoshkin, A.N. (2014) Genetic Algorithm with Fast Greedy Heuristic for Clustering and Location Problems. *Informatica*, Vol.38(3), pp.229-240.
- [22] Osman I. H., Laporte G. (1996) Metaheuristics: a bibliography. *Ann. Oper. Res.*, Vol.63. pp.513-628. <https://doi.org/10.1007/BF02125421>.
- [23] Mladenovic N., Hansen P. Variable neighborhood search. *Comput. Oper. Res.*, Vol.24, P.1097-1100. [https://doi.org/10.1016/S0305-0548\(97\)00031-2](https://doi.org/10.1016/S0305-0548(97)00031-2).
- [24] Hansen P., Mladenovic N. (2001) Variable neighborhood search: principles and applications (invited review). *European J. Oper. Res.*, Vol.130(3), pp.449-467. [https://doi.org/10.1016/S0377-2217\(00\)00100-4](https://doi.org/10.1016/S0377-2217(00)00100-4)
- [25] Garcia-Lopez F., Melian-Batista B., Moreno-Perez J.A., Moreno-Vega M. (2002) The parallel variable neighborhood search for the p-median problem. *Journal of Heuristics*, Vol.8, pp. 375-388 (2002). <https://doi.org/10.1023/A:1015013919497>.
- [26] Brimberg J., Mladenovic N. (1996) A variable neighborhood algorithm for solving the continuous location-allocation problem. *Stud. Locat. Anal.* V. 10. pp. 1-12.
- [27] Hansen P., Mladenovic N., Perez-Brito D. (2001) Variable neighborhood decomposition search. *J. Heuristics*, Vol.7 (4), pp. 335-350. <https://doi.org/10.1023/A:1011336210885>.
- [28] Goldberg D. E. (1989) *Genetic algorithms in search, optimization, and machine learning*. Reading, MA: Addison-Wesley. <https://doi.org/10.5555/534133>.
- [29] Houck C.R., Joines J. A., G.Kay. M. (1996) Comparison of Genetic Algorithms, Random Restart, and Two-Opt Switching for Solving Large Location-Allocation Problems. *Computers and Operations Research*, Vol. 23, pp. 587-596. [https://doi.org/10.1016/0305-0548\(95\)00063-1](https://doi.org/10.1016/0305-0548(95)00063-1).
- [30] Alp O., Erkut E., Drezner Z. (2003) An Efficient Genetic Algorithm for the p-Median Problem. *Annals of Operations Research*. Vol.122, pp.21-42, <https://doi.org/10.1023/A:1026130003508>. (2003).
- [31] Neema M.N., Maniruzzaman K.M., Ohgai A. (2011) New Genetic Algorithms Based Approaches to Continuous p-Median Problem. *Netw. Spat. Econ.*, Vol.11, pp.83-99, <https://doi.org/10.1007/s11067-008-9084-5>.
- [32] UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. access date 28.03.2019.
- [33] Clustering basic benchmark [<http://cs.joensuu.fi/sipu/datasets>], access date 28.03.2019.
- [34] Wilcoxon F. (1945) Individual comparisons by ranking methods. *Biometrics Bulletin*, Vol.1(6), pp. 80–83. <https://doi.org/10.2307/3001968>.
- [35] Derrac J., Garcia S., Molina D., Herrera F. (2011) A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, Vol. 1(1), pp. 3-18, <https://doi.org/j.swevo.2011.02.002>.
- [36] Sheng W., Liu X. (2006) A genetic k-medoids clustering algorithm. *Journal of Heuristics*. Vol.12, No.6. P. 447-466. <https://doi.org/10.1007/s10732-006-7284-z>.



# Hybrid Nearest Neighbors Ant Colony Optimization for Clustering Social Media Comments

Lucky Lucky and Abba Suganda Girsang

Computer Science Department, BINUS Graduate Program-Master of Computer Science

Bina Nusantara University, Jakarta, Indonesia 11480

E-mail: lucky@binus.ac.id, agirsang@binus.edu

**Keywords:** ant colony optimization, nearest neighbors, clustering, text mining, social media comments, twitter

**Received:** February 8, 2019

*Ant colony optimization (ACO) is one of robust algorithms for solving optimization problems, including clustering. However, high and redundant computation is needed to select the proper cluster for each object, especially when the data dimensionality is high, such as social media comments. Reducing the redundant computation may cut the execution time, but it can potentially decrease the quality of clustering. With the basic idea that nearby objects tend to be in the same cluster, the nearest neighbors method can be used to choose the appropriate cluster for some objects efficiently by considering their neighbor's cluster. Therefore, this paper proposes the combination of nearest neighbors and ant colony optimization for clustering (NNACOC) which can reduce the computation time but is still able to retain the quality of clustering. To evaluate its performance, NNACOC was tested using some benchmark datasets and twitter comments. Most of the experiments show that NNACOC outperformed the original ant colony optimization for clustering (ACOC) in quality and execution time. NNACOC also yielded a better result than k-means when clustering the twitter comments.*

*Povzetek: Predstavljen je hibridni algoritem najbližjih sosedov z optimizacijo mravljinčnih kolonij za grupiranje komentarjev socialnih medijev.*

## 1 Introduction

Nowadays, clustering plays an important role in many applications, such as business intelligence and analytic [1], public health and security [2], as well as the energy saving of internet of things [3], [4]. Clustering also has been implemented in many cases of text mining. According to [5], with the rapid growth of social media usage, petabytes of data had been generated; most of them are in the form of text, blogs, Twitter comments, Facebook feeds, chats, e-mails, and reviews. Therefore, clustering the social media comments has drawn many interests from government to businesses for reading people's opinions quickly and accurately. Some of the examples are detecting public concern about Ebola virus in United States of America using Twitter comments [6], and analyzing the engagement level of three largest pizzas chains with their customers through Facebook and Twitter comments [7].

In the study by [8], most of clustering methods can be considered as optimization problems for finding the most optimal data partitioning based on the objective function. One of the most popular clustering algorithms is k-means which was introduced in 1955 and is still widely used until now [8]. However, according to [9], [10], k-means algorithm sometimes fall into local optima. Therefore, some researchers proposed metaheuristic approach for solving clustering problems such as artificial bee colony (ABC) [11], [12] and ant colony optimization (ACO) [13]–[21].

## 1.1 Basic concept

### 1.1.1 ACO algorithm

ACO algorithm was proposed by Dorigo [22] for choosing the shortest path in the Traveling Salesman Problem (TSP). ACO algorithm simulates the behavior of ants when going away from their nest to the source of food and going back to the nest. Those ants use pheromone trail they drop on each travel to communicate and find the shortest path as illustrated in Figure 1.

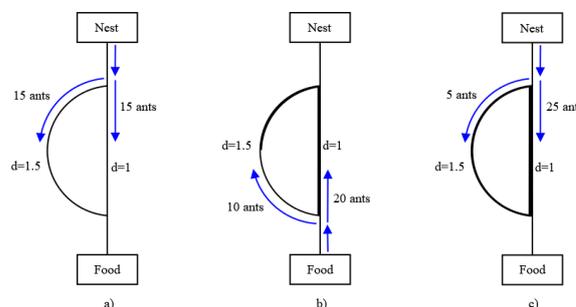


Figure 1: The basic principle of ACO algorithm.

There are two alternative routes between nest and food source; one has a shorter distance equals to 1 ( $d=1$ ) and the other one has a longer distance equals to 1.5 ( $d=1.5$ ). In part a, there is no pheromone on all paths. So, the probability that each ant chooses one of the two ways is equal. Part b shows the condition when ants travel back from food source to their nest. Since more ants can go

faster through the shorter path, the shorter path will have more pheromone than the other one. The pheromone on the longer path also becomes weaker because the evaporation occurs in each tour. Therefore, the shortest path has a bigger chance to be chosen. In part *c*, the number of ants choosing the shorter path increases as the pheromone level on it goes higher while the pheromone level on the longer path goes lower. As the cycle is repeated, all of the ants will eventually choose the shorter path.

The probability used by ants to choose their path is shown in (1).

$$\rho_{ij}^k = \begin{cases} \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{k \in allowed_k} [\tau_{ik}]^\alpha [\eta_{ik}]^\beta} & \text{if } j \in allowed_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

In (1),  $\tau_{ij}$  is the level of pheromone between node *i* and *j*. The  $\eta_{ij}$  is the heuristic information between node *i* and *j*. In the TSP case, it is the inverse distance between node *i* and *j*. The  $\alpha$  and  $\beta$  are the weight of importance for the pheromone level and heuristic information.

After generating the solution, the pheromone on each edge will be updated to improve the quality of the best solution found using (2).

$$\tau_{ij}(t) = (1 - \rho) \cdot \tau_{ij} + \sum_{k=1}^m \Delta\tau_{ij}^k \quad (2)$$

$$\Delta\tau_{ij}^k = \begin{cases} \frac{Q}{L_k} & \text{if } k\text{-th ant uses edge } (i, j) \text{ in its tour} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

In (2),  $\rho$  is the pheromone evaporation coefficient. Then in (3),  $\Delta\tau_{ij}^k$  is the pheromone deposited by *k*-th ant when walking through the node *i* to *j*, *Q* is the pheromone constant, and *L<sub>k</sub>* is the length of tour of the *k*-th ant.

### 1.1.2 ACO for clustering

ACO for Clustering (ACOC) was firstly introduced by Shelokar [13]. The basic idea of this technique is to represent the solution into a string containing cluster number assigned to each data. Table 1 shows an example of the solution string of a dataset with *N* = 8 object, *K* = 3 cluster number, and using *M* = 3 ants for constructing the solution.

SM	N							
	1	2	3	4	5	6	7	8
S1	2	2	1	1	3	3	3	3
S2	1	2	2	1	2	3	2	3
S3	1	1	1	2	2	3	3	3

Table 1: Illustration of solution strings in ACOC.

The basic principle of the solution construction in ACOC is almost similar to ACO, except in ACOC there is no heuristic information ( $\eta_{ij}$ ) used. There is only pheromone matrix from each object to each cluster for computing the possibility of each ant to choose a cluster that should belong to each object. The rule for choosing

cluster number is defined in (4) where *j* is the cluster number in *K* clusters.

$$\rho_{ij}^k = \frac{\tau_{ij}}{\sum_k \tau_{ik}}, j = 1 \dots K \quad (4)$$

The rule for pheromone evaporation is the same as (2) while the pheromone deposition rule is shown in (5).

$$\Delta\tau_{ij}^k = \frac{1}{F^k} \quad (5)$$

In (5),  $F^k$  is the fitness function of the solution generated by *k*-th ant. To obtain the fitness function, the centroids of each cluster in the generated solution must be calculated using the mean function. After that, the Sum of Squared Error (SSE) of Euclidean distances between each object and the centroid is calculated for measuring the clustering quality. The best solution should have the most minimal SSE of euclidean distances.

The fitness function formula is shown in (6) where *K* is the number of clusters, *N* is the number of objects, and *n* is the number of attributes of an object. The  $x_{iv}$  is the *v*<sup>th</sup> dimension value of *i*<sup>th</sup> object attribute and  $m_{jv}$  is the *v*<sup>th</sup> dimension value of the centroid for *j*<sup>th</sup> cluster. The  $w_{ij}$  is the weight that indicates if the *i*<sup>th</sup> object belongs to *j*<sup>th</sup> cluster. The value is 1 if the *i*<sup>th</sup> object belongs to *j*<sup>th</sup> cluster or 0 otherwise.

$$F(w, m) = \sum_{j=1}^K \sum_{i=1}^N \sum_{v=1}^n w_{ij} \|x_{iv} - m_{jv}\| \quad (6)$$

On each iteration, the pheromone matrix between each object and cluster is updated. The bigger the value of the pheromone between an object and certain cluster, the bigger the chance that the object will be assigned to that cluster.

ACOC also implements the elitist ant strategy, which means that only *n*-best ants or solutions will be permitted to deposit the pheromone. The value of *n* is usually 20% from the total number of ants. Besides that, ACOC also uses the local search to improve its generated solution. The local search, which is similar to mutation in Genetic Algorithm, is only performed on 20% of ants with the best fitness value. The process starts from generating *N* random numbers sequentially where *N* is the number of objects in a solution. If the generated random numbers were smaller than the pre-determined threshold parameter, the objects in the same sequence as those generated random numbers must change its cluster to a different one. After that, the fitness of the mutated solution will be calculated. If it had a better fitness, then it will replace the current solution. Otherwise, the current solution will be used.

## 1.2 Related works

According to [23], the ant algorithm for clustering can be divided into two groups, ant-based sorting and ACO based clustering. Ant-based sorting algorithm uses two dimensions grid (x, y) plane. In that algorithm, the objects are scattered randomly at first. After that, the artificial ants

will pick the object which is dissimilar to its neighborhood, move it to another location with similar objects, and then drop it. The studies [15], [18]–[20] also used this basic concept in their proposed solution. Although the ant-based sorting does not need a cluster number to be defined a priori, it needs post-processing to identify the generated clusters and requires high processing time [23]. That was proven in some studies where the cluster number should be analyzed visually after the clustering was done [19]. Besides that, the iteration number could reach 15000 for clustering Iris dataset [15].

Another ant algorithm for clustering is ACO based clustering which uses the same concept of solution string to represent the clustering solution as explained in section 1.1.2. The solution string is constructed on each iteration and evaluated by the objective function to find the most optimal one. Although the cluster number must be defined a priori, ACO based clustering is more efficient in computation than ant-based sorting. Also, it does not need post-processing after the clustering is done [23]. Aside of ACO, some of the proposed clustering algorithms also use the same concept of solution string as ACO based clustering [12], [24]–[26].

The first implementation of ACO based clustering is ACOC [13]. After that, ACOC has been improved in some studies such as [14] which modified the original ACOC by keeping the identified best solution as the initial solution for the next iteration and adding the capability to determine the optimal cluster number automatically using Jaccard index. However, the research shows that the algorithm spends more time to run. The research [16] takes a different approach by combining the ACOC with k-means algorithm. K-means is used to generate the initial solution to be explored later by ACO. However, the algorithm is only tested for processing the financial services data. Furthermore, the research [17] also uses the concept of ACO based clustering; however, its focus is for building the classification model based on the training dataset which is clustered using ACO.

The recent research proposes fast ant colony optimization for clustering (FACOC) to improve the efficiency of computation in ACOC [21]. FACOC uses a threshold value to determine if a cluster number became common for an object after it is being chosen for several times. If a cluster number for an object became common, on the next iteration, that cluster number for that object will be simply chosen without computing the probability anymore. This can cut the redundant computations, so that the execution time can be faster. Furthermore, the object with common cluster number will not be affected by local search. However, the result shows that FACOC outputs lower clustering quality than ACOC.

### 1.3 Problem definition

ACOC uses the probabilistic calculation for choosing the proper cluster for each object based on the strength of the pheromone. That calculation must be done for each object on each iteration. Because of that, the computation in ACOC is high, especially when it is used for clustering large datasets with high dimensionality, such as text and

social media comments. Even though the method for reducing the redundant computations has been proposed in FACOC, its performance shows the degradation of the clustering quality. Therefore, the problem that this research tries to solve is how to reduce the computation time of ACOC and retain the clustering quality at the same time.

### 1.4 The objective and contribution

This paper proposes NNACOC, the hybrid of nearest neighbors and ACOC algorithm which is more efficient than ACOC but still able to retain the clustering quality. To achieve that objective, NNACOC uses the nearest neighbors algorithm to construct the list of nearest neighbors of each object. That list enables the algorithm to assign the same cluster for the current object and its nearest neighbors at the same time. By doing that, the computation for choosing cluster number probabilistically for those nearby objects can be reduced.

The idea that the nearest neighbors can be used for retaining the clustering quality is based on studies which indicate that the good clusters should have the most minimal sum of the euclidean distance between the objects and their cluster's centroid [11]–[13], [21], [27]. Based on that, the objects within the same cluster must be neighboring and near to each other. Therefore, the list of nearest neighbors of each object can be used by the algorithm to assign the appropriate cluster for an object and retain the clustering quality.

The remainder of this paper is organized as follows. Section 2 presents the detail of the proposed algorithm and the description of the datasets for the evaluation. Section 3 discusses the evaluation and its result. Section 4 is about the discussion of the result. Finally, the conclusions and future works are presented in Section 5.

## 2 Materials and methods

### 2.1 *N*-nearest neighbors construction

The important part in NNACOC is the list of *n*-nearest neighbors for each object in dataset. The *n* is the pre-determined number of nearest neighbors of the current object which will possibly be assigned with the same cluster number as the current object. When using relatively small dataset, it is still fine to construct the *n*-nearest neighbors for each object one by one using the brute force method. However, when the dataset is large, using the brute force method for constructing the *n*-nearest neighbors list is not feasible because it can be very resource and time consuming in computation.

To overcome the problem, some techniques are introduced for improving the speed of *n*-nearest neighbors construction. One of them is the ball tree algorithm. According to [28], ball tree is an improvement of k-nearest neighbors for faster execution which can be used for handling high dimensional entities. Text clustering usually deals with large number of features vector, which means the dimensionality is high. Thus, the ball tree algorithm is

chosen for  $n$ -nearest neighbors construction in this research.

Before the  $n$ -nearest neighbors construction can be performed, the text data must be vectorized or transformed into a vector space model. One of the most common vectorizing methods is term frequency - inverse document frequency (TF-IDF). The term importance is based on its occurrence frequencies in a document (TF). Then, it is normalized or reduced by its occurrence frequencies across the document collection (IDF). The TF-IDF method is also used in some of text clustering studies for vectorizing the text document [18], [24]–[26].

## 2.2 The NNACOC algorithm

The outline of proposed NNACOC algorithm is presented as follows. To make it easier to understand, the pseudocode

in Figure 2 is explained in the following step by step illustration.

**Step 1.** At line 1, the  $N$  nearest neighbors list for each object is constructed. Where  $N$  is the number of nearest neighbors which will be assigned to the same cluster automatically when an object is assigned to a cluster.

Let assume that there are 8 objects,  $N = 2$ , and the constructed  $N$  nearest neighbors list is shown in Table 2

**Step 2.** The iteration for constructing solutions is started. Each ant visits all objects one by one and the content of affected objects is checked. If the current object is not in the affected objects list, the cluster for the current object is selected using (4) (line 8 – 10).

```

1.  construct N nearest neighbors list
2.
3.  while termination condition is not met
4.    initialize empty solution string and affected objects
5.
6.    foreach ant in all ants
7.      foreach object in all objects
8.        if object is not in affected objects
9.          assign cluster to current object using eq. (4)
10.         update solution string
11.
12.         calculate the probability to affect neighbors
13.         if should affect neighbors
14.           get N nearest neighbors of current object
15.           foreach neighbor object in N nearest neighbors
16.             assign cluster to neighbor object
17.             update solution string and affected objects
18.
19.         calculate fitness of solution string using eq. (6)
20.
21.     fetch L best solutions from solution string
22.
23.     foreach solution in L best solutions
24.       calculate local search probability
25.       if should do local search
26.         initialize empty new solution
27.
28.         get affected objects in solution
29.         foreach object in affected objects
30.           if object has more than one cluster
31.             choose new cluster randomly then assign to object
32.             update new solution string
33.
34.         if new solution string is not empty and better than ith solution
35.           replace current solution in L best solution with new solution string
36.
37.     evaporate pheromone using eq. (2)
38.
39.     foreach solution in L best solution
40.       deposit pheromone using eq. (5)
41.
42.  display the best solution

```

Figure 2: The pseudocode of NNACOC.

Let assume that the affected objects list is empty, the ant is on object A, and then the chosen cluster is 1. So, object A is assigned to cluster 1.

**Step 3.** After that, the probability calculation is done to decide if the cluster assignment in step 2 should affect the neighbors. It is done by randomizing the float number between 0 and 1. If the result is smaller or equal to pre-determined threshold, then the cluster assignment should affect its neighbors too (line 12 – 17).

Let assume that the cluster assignment should affect the neighbors. Based on the Table 2, C and E are the nearest neighbors of A. Therefore, C and E will be automatically assigned to cluster 1. If ant visit object C or E on the next iteration, then process in line 9 to 17 is skipped. This is how the redundant computations can be reduced.

When visiting each object, there is a possibility that the affected object is assigned to more than one cluster. For example, the ant is on object B then assigns it to cluster 2. The probability calculation indicates that the nearest neighbors should be affected to. Then, based on Table 2, D and E are assigned to cluster 2 as well. Object E has been assigned to cluster 1 previously. So, object E has the possibility to belong to cluster 1 or cluster 2. This condition can be described in Table 3.

**Step 4.** After each ant finished constructing the solution, some of the best solutions are selected using the same elitist ants strategy as [13] to be processed in local search (line 24 – 35). The same probability calculation as in step 2 is done to decide whether the local search should be done or not. The local search itself is done by randomly selecting other possible clusters to the object that has the possibility to belong to more than one cluster only. If the newly mutated solution has a better fitness than the current solution, it will replace the current solution.

For the example, based on Table 3, object E is currently assigned to cluster 1 because that cluster is chosen at the first time, but it is also possible to belong to cluster 2. To give a better visualization, Figure 3 illustrates the current solution in 2D representation.

The local search process mutated the current solution by replacing the cluster of object E with the different

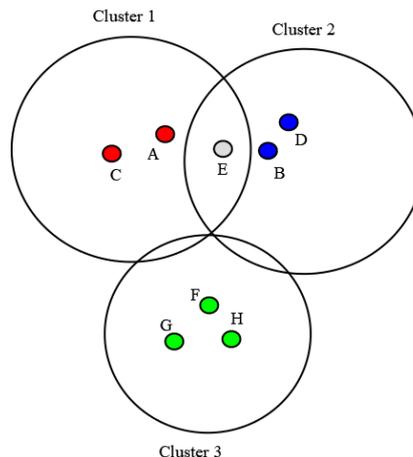


Figure 3: The 2D visualization of clustering solution.

cluster than the current one. In this case, it is cluster 2. If the candidate is more than one, it will be selected randomly. After that, the new solution is evaluated. If it is better than the current solution, then it will replace the current one. Otherwise, the current one is kept.

**Step 5.** Then, the pheromone evaporation is done to reduce the possibility of bad solution from being chosen (line 37). After that, pheromone deposition is done only for the pre-determined number of the best solutions to increase the possibility of choosing good solution (line 39 – 40).

**Step 6.** When the termination condition is met, such as reaching certain iteration or maximum execution time, the best solution found is displayed (line 42).

The block diagram of the proposed clustering system can be seen in Figure 4. It contains three main parts, which are dataset collection, nearest neighbors construction, and clustering using NNACOC. The dataset collection part is divided into two parts, one for collecting the benchmark dataset and the other one for collecting Twitter comments dataset which contains data collection process, and data cleansing and pre-processing.

**2.3 Fitness function**

For measuring the clustering quality on numerical dataset, NNACOC uses the same fitness function as ACOC and FACOC which is defined in (6). However, for measuring clustering quality on social media comments, NNACOC uses the sum of cosine distance instead of SSE of euclidean distance between comments and their cluster centroids. The cosine distance can be defined as 1 – cosine similarity. The equation for cosine similarity is defined in (7), where X and Y are the vectorized texts to be compared,  $x_i$  and  $y_i$  are the words vector or bag of words from X and Y.

$$\cos(X, Y) = \frac{\sum_i x_i \cdot y_i}{\sqrt{\sum_i (x_i)^2} \cdot \sqrt{\sum_i (y_i)^2}} \tag{7}$$

The cosine similarity itself is the most common method for calculating the similarity between texts that

Object	Nearest Neighbors
A	C, E
B	D, E
C	A
D	B
E	A, B
F	G, H
G	F, H
H	F, G

Table 2: List of the nearest neighbors for each object.

Object	Clusters
C	1
D	2
E	1,2

Table 3: The list affected objects with its possible clusters.

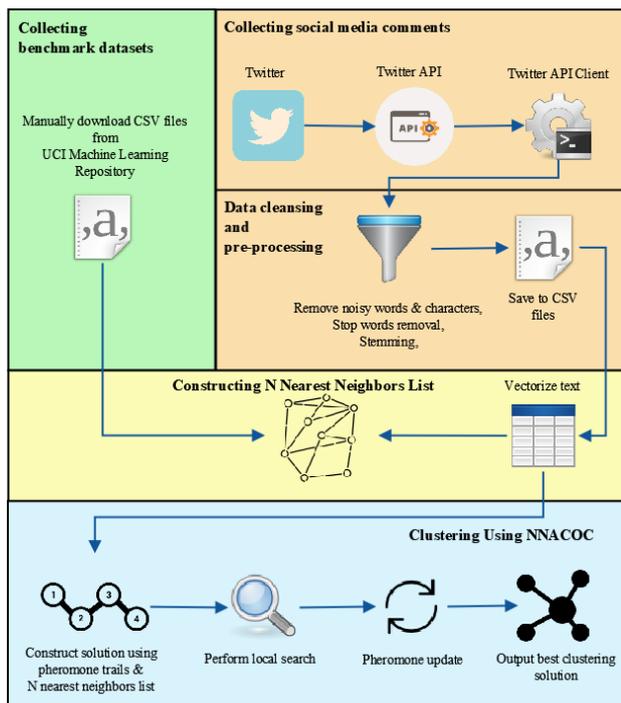


Figure 4: Block diagram of the proposed system.

has been previously converted into vector-space model. Cosine similarity is also used in some text clustering studies for evaluating the clustering quality by calculating the similarity between the vectorized text and its cluster’s centroid [18], [24], [25].

### 2.4 Dataset collecting

There are two kinds of datasets used in this research. One is for evaluating the standard clustering and the other one is for evaluating the social media comments clustering. Four benchmark datasets for standard clustering are collected from [29] as shown in Table 4.

For text clustering, the Twitter comments in Bahasa Indonesia, which were collected using certain hashtags (#politik, #keuangan, #teknologi, #traveling, #kesehatan, #kuliner, and #olahraga), are populated into four different datasets. Then, because of the noisy characteristics of social media comments, the dataset is populated and pre-processed using the steps illustrated in Figure 4, in the section of collecting social media comments dataset.

The steps in that section can be described as follows.

**Step 1.** In data collecting section, the Twitter Search API is used by the Twitter API client script for retrieving the comments with certain hashtags.

**Step 2.** After that, in data cleaning and pre-processing section, the collected comments are cleaned using regular expression from noisy words and characters such as hashtags, mention, URL, emoticons, and repeating characters in a word, as shown in Table 5. Then, the comments are normalized using the program for stop words removal [30] and stemming [31]. Finally, the data is saved into a CSV file.

The specification of Twitter comments datasets is shown in Table 6. For each dataset, the number of hashtags is assumed as the number of clusters. Each of them also has number of attributes, which is the total of feature vectors generated by the TF-IDF algorithm.

## 3 Evaluation and results

### 3.1 Evaluation environment

For the evaluation, ACOC and NNACOC are implemented in Python 3 programming language. The k-means algorithm is used as the additional comparison which is also implemented using Python 3. Those algorithms are tested in a laptop with Intel i3 processor, 4 gigabytes RAM, and Arch Linux as the operating system.

### 3.2 Parameter settings

Some of the parameters used in this research are the same for ACOC and NNACOC, as shown in Table 7, while some are specific for NNACOC only. Others are shown in Table 8.

We use the elitist ant strategy, which according to [13], the ideal value for  $e$  is about 20% of  $m$ . As explained in section 1.1.2 on the basic concept of ACOC, only the elitist ants ( $m$  ants with best solution) are permitted to deposit pheromone. Then, only their solutions will be used in local search. Furthermore, the  $p_s$  value is used to control the probability of the mutation in local search process.

Dataset Name	Number of Clusters	Number of Samples	Number of Attributes
Iris	3	150	4
Wine	3	178	13
Breast Cancer (Wisconsin)	2	699	9
Contraceptive Method Choice	3	1473	9

Table 4: Benchmark datasets for clustering.

Regex Pattern	Target
(?:\#[\w_]+[\w'_-]*[\w_]+)	Hashtag
(?:@[w_]+)	Mention
http[s]?://(?:[a-z][0-9] [\$- _@.&+![*\(\),](?:%[0-9a-f][0-9a-f])]+)	URL
(?:[=:;][oO\-\]?[D\]\(\)\[\]\OpP])	Emoticon
([a-z])\1+	Repeating characters in a word

Table 5: Regex pattern for cleaning the comments.

Dataset Name	Hashtags in Dataset	Total Tweets	Total Feature Vectors
111 tweets with 3 hashtags	#politik (41 tweets) #teknologi (37 tweets) #olahraga (33 tweets)	111	453
522 tweets with 3 hashtags	#politik (205 tweets) #teknologi (153 tweets) #olahraga (164 tweets)	522	1511
738 tweets with 5 hashtags	#politik (167 tweets) #keuangan (149 tweets) #teknologi (127 tweets) #kesehatan (170 tweets) #olahraga (125 tweets)	738	1877
1013 tweets with 7 hashtags	#politik (167 tweets) #keuangan (149 tweets) #teknologi (127 tweets) #traveling (103 tweets) #kesehatan (170 tweets) #kuliner (172 tweets) #olahraga (125 tweets)	1013	2545

Table 6: Datasets for Twitter comments clustering.

Dataset	ACOC		NNACOC		
	Avg. Performance	Avg. Execution Time (s)	Avg. Performance	Avg. Execution Time (s)	Nearest Neighbors Construction Time (s)
Iris	79.01	11.15	<b>78.94</b>	<b>8.88</b>	0.31
Wine	2370689.69	14.09	2370689.69	<b>10.91</b>	0.24
Breast Cancer Wisconsin	19516.14	42.85	<b>19444.15</b>	<b>32.43</b>	0.31
Contraceptive Method Choice	36458.68	99.56	<b>26322.97</b>	<b>76.59</b>	0.33

Table 7: The average of performance and execution time between ACOC and NNACOC on numerical datasets.

The value of  $nn$  decides how many neighbors can be affected after the cluster number assignment of the current object. We found that 20 is the most optimal value despite the total number of objects. Setting the value of  $nn$  too low can slow down the speed, while setting it too high can degrade the performance. The parameter  $q1$  controls the probability if the cluster number assignment to an object also affects its neighbors. The purpose of  $q1$  is to make the solution construction more dynamic, which means that the cluster number assignment to an object may, but not always, affect its neighbors.

Because of the inherent random behaviour of ACO, the evaluation is done in 10 trials for each dataset and algorithm. In each trial, the maximum iteration number for clustering the numerical datasets is set to 2000. Meanwhile, for the Twitter comments datasets the maximum iteration number is set to 3000 because the number of attributes of the vectorized Twitter comments

is much bigger. So, it is assumed that more iteration is needed to achieve the optimal result.

### 3.3 Results

The result of the average performance and execution time between ACOC and NNACOC for numerical datasets can be seen in Table 9 and the result for Twitter comments datasets can be seen in Table 10. As previously explained, the clustering performance is measured using SSE of euclidean distance for numerical dataset and sum of cosine distance for Twitter comments dataset.

As the additional performance comparison, k-means clustering is also applied to the datasets in Table 9. The result can be seen in the following list.

- Iris: 78.94
- Wine: 2370689.69
- Breast Cancer Wisconsin: 19323.17
- Contraceptive Method Choice: 23691.19

Parameter Name	Value
Number of ants ( $m$ )	25
Number of elitist ants ( $e$ )	5
Pheromone evaporation rate ( $\rho$ )	0.1
Local search probability ( $p_{ls}$ )	0.01

Table 8: Parameter settings for ACOC and NNACOC.

Parameter Name	Value
Nearest neighbors for each object ( $nn$ )	20
Probability to assign nearest neighbors with the same cluster ( $q1$ )	0.3

Table 9: Parameter settings specific to NNACOC.

Dataset	ACOC		NNACOC		
	Avg. Performance	Avg. Execution Time (s)	Avg. Performance	Avg. Execution Time (s)	Nearest Neighbors Construction Time (s)
111 tweets with 3 hashtags	80.35	34.56	<b>80.22</b>	<b>29.59</b>	0.03
522 tweets with 3 hashtags	407.92	403.72	<b>407.12</b>	<b>317.79</b>	1.28
738 tweets with 5 hashtags	582.21	638.20	<b>581.85</b>	<b>505.36</b>	1.45
1013 tweets with 7 hashtags	817.39	1061.17	<b>813.24</b>	<b>826.15</b>	3.79

Table 10: The average of performance and execution time between ACOC and NNACOC on Twitter comments datasets.

The dataset in Table 10 is also tested using k-means as the additional performance comparison. The result can be seen in the following list.

- 111 tweets with 3 hashtags: 82.8
- 522 tweets with 3 hashtags: 412.67
- 738 tweets with 5 hashtags: 586.69
- 1013 tweets with 7 hashtags: 825.86

### 3.4 Discussion

The results in the previous section indicate that NNACOC is not only faster than ACOC, but also able to output the equal or even better quality of clustering result than ACOC in most trials. Even though its quality loses to k-means in some numerical data clustering, NNACOC outperforms k-means when clustering the text datasets which have a more complex pattern. The original ACOC also performs better than k-means in text clustering but it loses to NNACOC in all trials.

We also observe how the optimal result is progressing on every iteration, both for ACOC and NNACOC. We display the charts of the iteration progress for each dataset in Figure 5 to Figure 12.

Based on the iterations progress charts for each dataset in Figure 5 to Figure 12, we found that NNACOC always starts with a better solution than ACOC. With that better start, NNACOC has a bigger chance to reach the optimal solution quicker in less iteration than ACOC.

The success key of NNACOC is certainly the nearest neighbors list, which is used in solution construction process. It can reduce the redundant probabilistic calculations by automatically assigning the same cluster to  $n$ -nearest neighbors of certain objects. Because the objects in the same clusters are usually near to each other, the cluster assignment can be done appropriately, so that the clustering quality can be ensured. The local search process also plays an important role in selecting the most suitable cluster if an object has the possibility to belong to more than one cluster.

However, the automatic cluster assignment should not always happen. The reason is to avoid that the solution falls into local optimum, by giving a chance for the algorithm to explore better possible solutions. So, we set 30% for its probability to happen. The number of how many nearest neighbors can be automatically assigned to a cluster also needs to be set carefully. Too low value can slow down the speed because the redundant probabilistic calculations will be increased. Too high value potentially includes many inappropriate objects which should belong to different cluster. So, it can degrade the clustering quality. We found that 20 is the most optimal number without being affected by the total of objects.

In Table 9 and Table 10, we can also see that the nearest neighbors construction process at the very beginning run of the NNACOC algorithm needs extra time than the ACOC. However, when that extra time is added to the average of NNACOC execution time, the total of

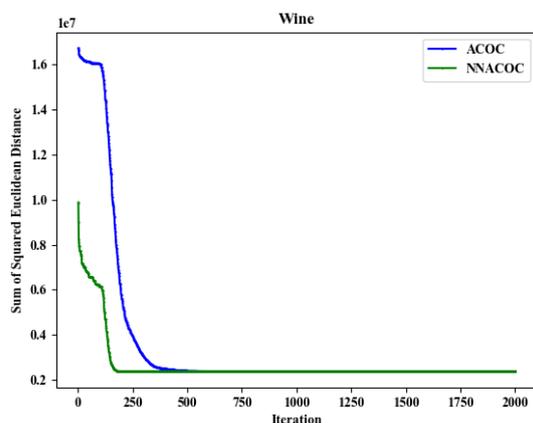


Figure 5: Clustering progress on Wine dataset.

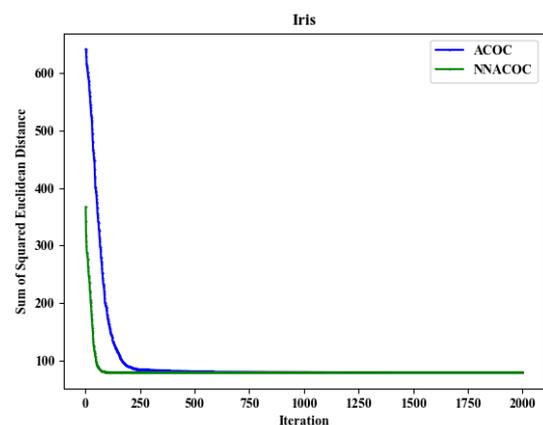


Figure 6: Clustering progress on Iris dataset.

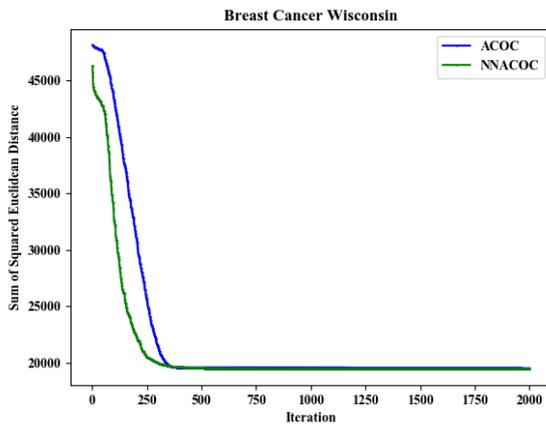


Figure 7: Clustering progress on Breast Cancer Wisconsin dataset.

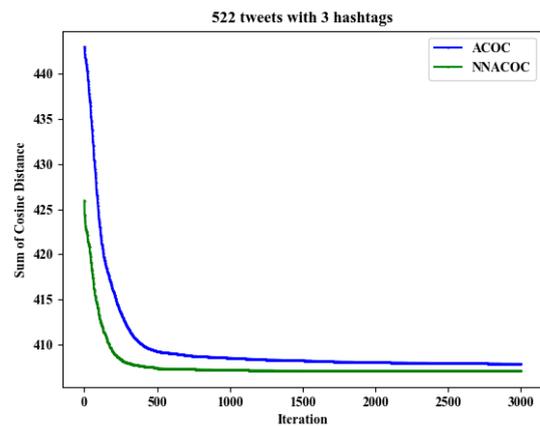


Figure 10: Clustering progress on “522 tweets 3 with hashtags” dataset.

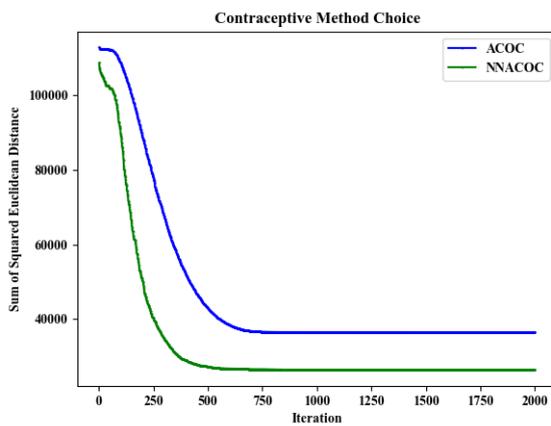


Figure 8: Clustering progress on Contraceptive Method Choice dataset.

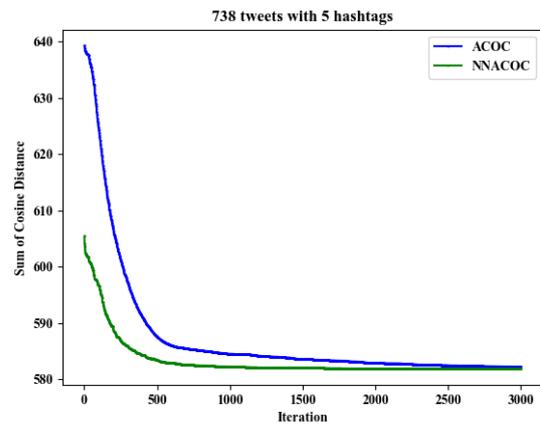


Figure 11: Clustering progress on “738 tweets with 5 hashtags” dataset.

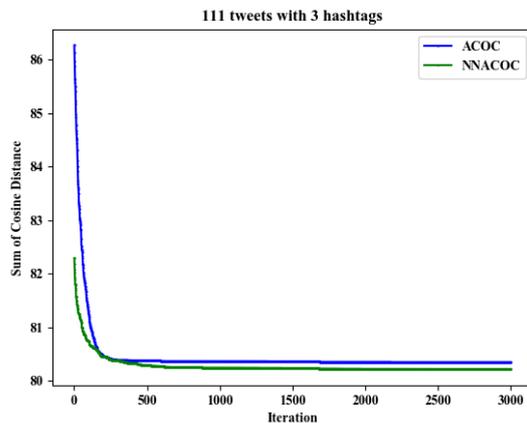


Figure 9: Clustering progress on “111 tweets 3 with hashtags” dataset.

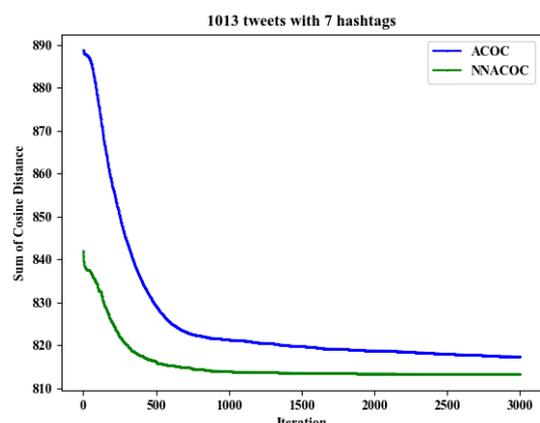


Figure 12: Clustering progress on “1013 tweets with 7 hashtags” dataset.

execution time is still faster than the average execution time of ACOC in all test cases. Moreover, according to our experiments, the extra times needed to construct the nearest neighbors range from 0.03 seconds to 3.79 seconds depending on the size and complexity of the datasets. But, NNACOC can reduce more execution times ranging from 2.62 seconds to 235.02 seconds. The details are shown in Table 11 and Table 12.

To give a better visualization of the execution time comparison between ACOC and NNACOC, we present the following chart in Figure 13 and 14.

Based on the charts, we can see that the difference of total execution time between ACOC and NNACOC becomes bigger when the dataset gets larger. There is also a blue bar at the bottom of each NNACOC chart for showing the nearest neighbors construction time in NNACOC. However, its appearance is almost

Dataset	Nearest Neighbors Construction Time (s)	Reduced Clustering Time (s)
Iris	0.31	2.62
Wine	0.24	3.18
Breast Cancer Wisconsin	0.31	10.42
Contraceptive Method Choice	0.33	22.97

Table 11: The additional time for nearest neighbors construction versus the reduced clustering time in NNACOC on numerical datasets.

Dataset	Nearest Neighbors Construction Time (s)	Reduced Clustering Time (s)
111 tweets with 3 hashtags	0.03	4.97
522 tweets with 3 hashtags	1.28	85.93
738 tweets with 5 hashtags	1.45	132.84
1013 tweets with 7 hashtags	3.79	235.02

Table 12: The additional time for nearest neighbors construction versus the reduced clustering time in NNACOC on Twitter comments datasets.

unnoticeable. This is a good sign which means that the additional time for the nearest neighbors construction is not significant compared to the reduced execution time.

### 4 Conclusion

Based on the evaluation result of NNACOC algorithm, it can be concluded that the nearest neighbors algorithm can be used to improve the ACO based algorithm for clustering, especially when clustering the large datasets. With the help of nearest neighbors information for each object, the clustering using ACO can be done faster and more efficient without sacrificing the performance. The evaluation results also show that both ACOC and NNACOC are able to output a better clustering quality than k-means when tested against text dataset. However, even though NNACOC is faster than ACOC and performs better than k-means in text clustering, it still cannot match the speed of k-means in the same case. So, the next challenge for future research is to investigate how to speed up the ACO based clustering algorithm to have the same speed as k-means, or at least close to it, without losing its ability to retain the clustering quality.

### 5 References

[1] S. Rajagopal, "Customer data clustering using data mining technique," *International Journal of Database Management Systems*, vol. 3, no. 4, 2011. <https://doi.org/10.5121/ijdms.2011.3401>

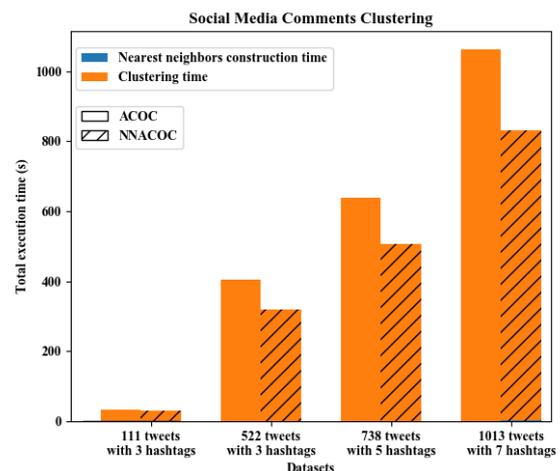


Figure 13: The execution time of ACOC and NNACOC on Twitter comments datasets.

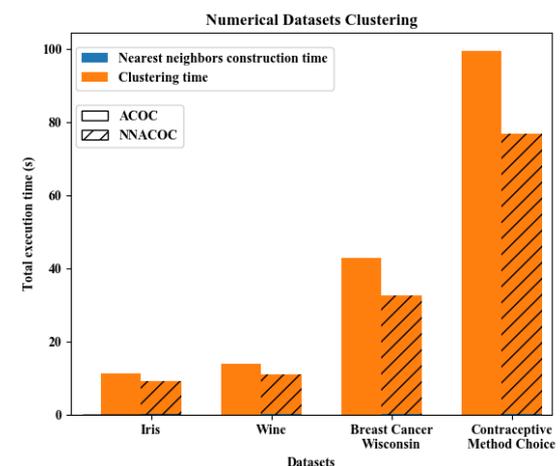


Figure 14: The execution time of ACOC and NNACOC on numerical datasets.

[2] H. Chen, R. H. L. Chiang, and V. C. Storey, "Business intelligence and analytics: from big data to big impact," *MIS Quarterly*, vol. 36, no. 4, pp. 1165–1188, 2012. <https://doi.org/10.2307/41703503>

[3] H. Chen, Z. Lv, R. Tang, and Y. Tao, "Clustering Energy-Efficient Transmission Protocol for Wireless Sensor Networks Based on Ant Colony Path Optimization," in *2017 International Conference on Computer, Information and Telecommunication Systems (CITS)*, 2017. <https://doi.org/10.1109/cits.2017.8035280>

[4] M. Abdelhafidh, M. Fourati, L. C. Fourati, A. Ben Mnaouer, and M. Zid, "Linear WSN Lifetime Maximization for Pipeline Monitoring using Hybrid K-means ACO Clustering Algorithm," in *2018 Wireless Day (WD)*, 2018. <https://doi.org/10.1109/wd.2018.8361715>

[5] N. Hardeniya, J. Perkins, D. Chopra, N. Joshi, and M. Iti, *Natural Language Processing: Python and NLTK*. Birmingham: Packt Publishing, 2016.

[6] A. J. Lazard, E. Scheinfeld, J. M. Bernhardt, G. B. Wilcox, and M. Suran, "Detecting themes of public concern: A text mining analysis of the Centers

- for Disease Control and Prevention's Ebola live Twitter chat," *American Journal of Infection Control*, 2015  
<https://doi.org/10.1016/j.ajic.2015.05.025>
- [7] W. He, S. Zha, and L. Li, "Social media competitive analysis and text mining: A case study in the pizza industry," *International Journal of Information Management*, vol. 33, pp. 464–472, 2013.  
<https://doi.org/10.1016/j.ijinfomgt.2013.01.001>
- [8] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010  
<https://doi.org/10.1016/j.patrec.2009.09.011>
- [9] Y. Kao and K. Cheng, "An ACO-Based Clustering Algorithm," in *ANTS 2006: Ant Colony Optimization and Swarm Intelligence*, 2006, pp. 340–347.  
[https://doi.org/10.1007/11839088\\_31](https://doi.org/10.1007/11839088_31)
- [10] B. C. Mohan and R. Baskaran, "A survey: Ant Colony Optimization based recent research and implementation on several engineering domain," *Expert Systems with Applications*, vol. 39, pp. 4618–4627, 2012  
<https://doi.org/10.1016/j.eswa.2011.09.076>
- [11] F. Zabihi and B. Nasiri, "A Novel History-driven Artificial Bee Colony Algorithm for Data Clustering," *Applied Soft Computing*, vol. 71, 2018.  
<https://doi.org/10.1016/j.asoc.2018.06.013>
- [12] A. S. Girsang, Y. Mulyono, and F. Fanny, "Fast Artificial Bee Colony for Clustering," *Informatica*, vol. 42, no. 2, 2018.
- [13] P. S. Shelokar, V. K. Jayaraman, and B. D. Kulkarni, "An Ant Colony Approach for Clustering," *Analytica Chimica Acta*, vol. 509, no. 2, pp. 187–195, 2004. <https://doi.org/10.1016/j.aca.2003.12.032>
- [14] X. Liu and H. Fu, "An Effective Clustering Algorithm With Ant Colony," *Journal of Computers*, vol. 5, no. 4, pp. 598–605, 2010.  
<https://doi.org/10.4304/jcp.5.4.598-605>
- [15] W. Gao, "Improved Ant Colony Clustering Algorithm and Its Performance Study," *Computational Intelligence and Neuroscience*, vol. 2016, no. 19, 2016  
<https://doi.org/10.1155/2016/4835932>
- [16] D. Fogarty, A. George, and S. N. Bhaduri, "New Methods in Ant Colony Optimization Using Multiple Foraging Approach to Increase Stability," in *Advanced Business Analytics*, Singapore: Springer, 2016, pp. 131–138  
[https://doi.org/10.1007/978-981-10-0727-9\\_10](https://doi.org/10.1007/978-981-10-0727-9_10)
- [17] K. M. Salama and A. M. Abdelbar, "Learning cluster-based classification systems with ant colony optimization algorithms," *Swarm Intelligence*, vol. 11, no. 3–4, 2017. <https://doi.org/10.1007/s11721-017-0138-5>
- [18] H. Pang, H. Zhao, W. Li, and Y. Ma, "The Research on the Improved Ant Colony Text Clustering Algorithm," in *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, 2017.  
<https://doi.org/10.1109/icbda.2017.8078833>
- [19] T. M. Pacheco, L. B. Gonçalves, V. Ströele, and S. S. R. F. Soares, "An Ant Colony Optimization for Automatic Data Clustering Problem," in *2018 IEEE Congress on Evolutionary Computation (CEC)*, 2018. <https://doi.org/10.1109/cec.2018.8477806>
- [20] C. Fahy, S. Yang, and M. Gongora, "Ant Colony Stream Clustering: A Fast Density Clustering Algorithm for Dynamic Data Streams," in *IEEE Transactions on Cybernetics*, 2018  
<https://doi.org/10.1109/tcyb.2018.2822552>
- [21] A. S. Girsang, W. T. Cenggoro, and K.-W. Huang, "Fast Ant Colony Optimization for Clustering," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 1, 2018.  
<https://doi.org/10.11591/ijeecs.v12.i1.pp78-86>
- [22] M. Dorigo, V. Maniezzo, and A. Colnari, "The Ant System: Optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics—Part B*, vol. 26, no. 1, pp. 1–13, 1996.  
<https://doi.org/10.1109/3477.484436>
- [23] A. M. Jabbar, K. R. Ku-Mahamud, and R. Sagban, "Ant-based sorting and ACO-based clustering approaches: A review," in *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, 2018  
<https://doi.org/10.1109/iscaie.2018.8405473>
- [24] L. M. Abualigah and A. T. Khader, "Unsupervised text feature selection technique based on hybrid particle swarm optimization algorithm with genetic operators for the text clustering," *The Journal of Supercomputing*, vol. 73, no. 11, 2017.  
<https://doi.org/10.1007/s11227-017-2046-2>
- [25] L. M. Abualigah, A. T. Khader, and E. S. Hanandeh, "Hybrid clustering analysis using improved krill herd algorithm," *Applied Intelligence*, vol. 48, no. 11, 2018. <https://doi.org/10.1007/s10489-018-1190-6>
- [26] N. Kushwaha and M. Pant, "Link based BPSO for feature selection in big data text clustering," *Future Generation Computer Systems*, vol. 82, 2018.  
<https://doi.org/10.1016/j.future.2017.12.005>
- [27] E. Queiroga, A. Subramanian, and L. dos A. F. Cabral, "Continuous Greedy Randomized Adaptive Search Procedure for data clustering," *Applied Soft Computing*, vol. 72, 2018  
<https://doi.org/10.1016/j.asoc.2018.07.031>
- [28] N. Bhatia and Vandana, "Survey of Nearest Neighbor Techniques," *International Journal of Computer Science and Information Security*, vol. 8, no. 2, pp. 302–305, 2010.
- [29] "UCI Machine Learning Repository." [Online]. Available: <https://archive.ics.uci.edu/ml/index.php>. [Accessed: 30-Sep-2018].
- [30] GitHub Inc., "GitHub - masdevid/ID-Stopwords: Stopwords collection of Bahasa Indonesia collected from many sources." [Online]. Available: <https://github.com/masdevid/ID-Stopwords>. [Accessed: 30-Sep-2018].
- [31] GitHub Inc., "GitHub - sastrawi/sastrawi: High quality stemmer library for Indonesian Language (Bahasa)." [Online]. Available: <https://github.com/sastrawi/sastrawi>. [Accessed: 30-Sep-2018].



# A Robust Image Watermarking Scheme Based on the Laplacian Pyramid Transform

Nguyen Chi Sy, Ha Hoang Kha

Faculty of Electrical & Electronics Engineering, Ho Chi Minh City University of Technology, VNU-HCM, Vietnam

E-mail: chisy.nguyen@gmail.com, hkhkha@hcmut.edu.vn

Nguyen Minh Hoang

Saigon Institute of Information Communication Technology

E-mail: nmhoang@gmail.com

**Keywords:** robust watermarking, Laplacian pyramid, framing pyramids

**Received:** November 20, 2017

*This paper is concerned with the digital image watermarking techniques to protect intellectual property and to authenticate digital images. Different from the most conventional methods using the discrete cosine transforms (DCT) and discrete-wavelet transforms (DWTs), this paper exploits the improved Laplacian pyramid transform to develop a new image watermarking scheme in which the improved Laplacian pyramid transform is used to decompose and reconstruct the host image. Then, to select an appropriate watermarking solution, we investigate the various frequency sub-band regions with different the levels and strength factors to perform the watermark embedding. Finally, we conduct experiments to investigate the invisibility and robustness of the proposed algorithm in terms the peak signal-to-noise ratio (PSNR), normalized correlation (NC), and structural similarity index (SSIM). Experimental results showed that our proposed scheme offers good robustness and invisibility. As compared to the watermarking schemes using the curvelets, our watermarking scheme is more robust for the lossy JPEG compression and Gaussian low pass filtering attacks. In addition, our method is also efficient in terms of computational time.*

*Povzetek: Članek predstavlja novo obliko zaščite slik s pomočjo Laplacove piramidne transformacije.*

## 1 Introduction

Since the rapid development of communication networks and advances in digital signal processing have lead to the multimedia piracy issues, copyright protection of multimedia products has become an extensive research topic. To protect content of the multimedia data from the modification and to provide content authentication, watermarking methods have been used [1, 2, 3]. The watermarking method is to embed or hide digital information, known as watermark, into a multimedia product. Then, one can extract the watermark data when necessary for verifying the authenticity or the integrity of the carrier signals, identifying its owners, or tracing copyright infringements. The digital watermarking schemes can be applied to various digital multimedia data such as audio, image and video. In this paper, we focus on the digital watermarking for digital image.

### 1.1 Related works

Watermarking methods for digital images can be implemented in spatial domain or in transform domains. Watermarking schemes in spatial domain directly modify the gray level values of pixels. It has been known that the watermarking methods in spatial domain are ineffective since the watermarks can be easily destroyed by common sig-

nal processing operations [4]. To overcome this drawback, transform domain based watermarking schemes have been actively studied [5]. With regards to transform domain based watermarking schemes, two typical transforms that have been widely used are discrete cosine transform (DCT) and discrete wavelet transform (DWT) (see, for example, [3, 6, 7, 8, 9], and references therein). In general, the desired properties of watermarking schemes are the robustness, the invisibility and the capacity [10]. However, there are tradeoffs between these desirable properties. Reference [3] showed that the DCT based watermarking techniques are superior to ones based on spatial domain in terms of robustness. In addition, reference [6] demonstrated that DCT based watermarking schemes are robust against such the common signal processing attacks as low-pass filtering, reducing image quality (blurring), and adjusting contrast and brightness. However, DCT based watermarking techniques are unsustainable with the geometric transform attacks, for example, rotation, rescaling, and cutting operations [9]. Alternatively, by using the wavelet transform into watermarking schemes, the authors in [3] showed that watermarking schemes based on wavelet transform outperform those on DCT approaches. It should be noted that in compression and denoising applications, the coefficients in the transform domain are quantized or performed thresholding operations and, thus, there exist errors in reconstructed images.

Recently, the directional transforms have been exploited in the watermarking schemes. In [11], the authors introduced a digital image watermarking scheme using the curvelet transform domain. By using the scale distribution, Human Visual System (HVS) and curvelet coefficients, they selected the appropriate positions to insert the watermark. In their method, the binary watermark of  $21 \times 21$  was used. By experimental results, they showed that the embedding watermark in the curvelet domain ensures robustness and invisibility. In addition, they also indicated that watermarking in the curvelet domain offers the improved robustness and invisibility as compared to those in the ridgelet domain.

On the other hand, reference [12] proposed a digital image watermarking algorithm operating in the fast curvelet transform in which they selected the medium frequency coefficients to embed the binary watermark image of  $32 \times 32$  pixels. In [12], the authors illustrated that their proposed watermarking scheme is good at both invisibility and security. In addition, experiment results therein showed that their watermarking scheme offers good robustness against noise, cropping, filtering, JPEG compression and other attacks. Reference [13] proposed a blind watermarking based on the curvelet transform domain. In order to achieve both invisibility and robustness, many different scales of curvelet transform domain have been investigated to choose the appropriate scales to embed the watermark. Experimental results showed the advantages of their method as compared to a watermarking scheme in the DCT-DWT combined domain for the lossy JPEG compression attacks, speckle and Gaussian noise. Alternatively, the authors in [14] proposed Laplacian pyramid (LP) scheme to represent multi-resolution for images. The advantages of the LP scheme are its simplicity and low computation complexity. However, there exists some drawbacks in the LP schemes, such as implicit oversampling [15]. The authors in [15] proposed an improved Laplacian pyramid (LP) scheme by exploiting an efficient filter bank (FB). This approach is proved to be more efficient than the conventional methods for the signal reconstruction degraded by noise.

## 1.2 Motivation and contributions

Motivated from the advantages of an improved Laplacian pyramid (LP) scheme in [15] and inspired by the works in [11, 12, 16], we develop a blind watermarking algorithm in improved LP domain in which the symmetric bi-orthogonal and the new reconstruction methods are used. More specifically, we propose a blind watermarking using the improved Laplacian Pyramid transform. To balance the invisibility and the robustness, we exploit the low frequency and the mid frequency regions to embed the watermark. We investigate the various levels and strength factors to choose the appropriate values. The watermark is a binary image whose size is  $32 \times 32$ . To evaluate the performance of the watermarking schemes, we use the performance met-

rics as the peak signal-to-noise ratio (PSNR), normalized correlation (NC) and the structural similarity index (SSIM) to measure the invisibility and the robustness of the algorithms. Our experimental results showed that the proposed watermarking scheme offers high invisibility and robustness. As compared to the watermarking schemes based on curvelets, the proposed algorithm has better invisibility and robustness for the lossy JPEG compression attack.

The rest of the paper is organized as follows. In Section 2, we introduce a proposed watermarking scheme in which the improved Laplacian pyramid and a new reconstruction using projection are used. Then, watermark embedding and extracting schemes with selective levels and strength factors are introduced as well. Section 3 presents experimental results and discussions. Finally, the concluding remarks are presented in Section 4.

The contributions in this paper have been partly presented in the 2017 International Conference on Recent Advances in Signal Processing, Telecommunications & Computing [17].

## 2 Proposed watermarking scheme using Laplacian pyramid transform

In this section, we present a blind watermarking scheme. The embedding and extracting watermark algorithms are shown in Figure 1 and Figure 2, respectively. In our blind watermarking scheme, the host image is firstly analyzed into improved LP coefficients by Laplacian pyramid toolbox. We present the improved LP transform in detail in 2.1. To enhance the security for the watermark, we use the Arnold transform on the watermarks which shall be described in 2.2. The embedding scheme in detail is explained in 2.3 and the extracting scheme in detail is described in 2.4.

### 2.1 Laplacian pyramid and novel reconstruction method

#### 2.1.1 Burt and Adelson's Laplacian Pyramid

The block diagram for analysis and synthesis of the LP is shown in Figure 3 in which  $\mathbf{x}$  is the input signal, output  $\mathbf{c}$  is a coarse approximation while output  $\mathbf{d}$  is a difference between the original signal and the prediction  $\mathbf{p}$  [14, 15]. First, using low-pass filtering and down sampling yields a coarse approximation of the original. The coarse approximation signal is given by

$$c[n] = \sum_{k \in Z^d} x[k] h[Mn - k] = \langle x, \tilde{h}[\cdot - Mn] \rangle \quad (1)$$

where  $n, k \in Z^d$ ,  $h[n] = h[-n]$ . The coarse components are up-sampled and filtered to yield the prediction compo-

ment which is given by

$$p[n] = \sum_{k \in Z^d} c[k] g[n - Mk]. \quad (2)$$

In terms of matrices and vectors, the coarse and prediction components are expressed as  $\mathbf{c} = H\mathbf{x}$  and  $\mathbf{p} = G\mathbf{c}$  where  $\mathbf{x} = (x[n]:n \in Z^d)$ ,  $G$  and  $H$  correspond to  $G(\uparrow M)$  and  $H(\downarrow M)$ , respectively. Then, the difference between the

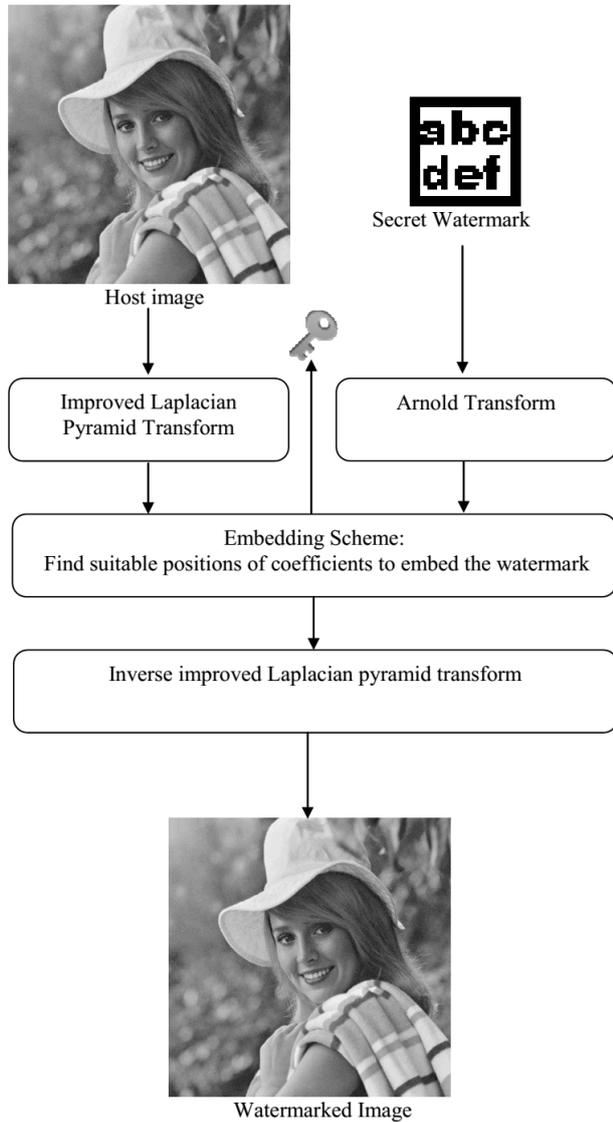


Figure 1: The proposed watermark embedding scheme

original signal and this predicted counterpart, known as the prediction error, is defined by

$$\mathbf{d} = \mathbf{x} - \mathbf{p} = \mathbf{x} - GH\mathbf{x} = (I - GH)\mathbf{x}. \quad (3)$$

Accordingly, we can rewrite the analysis operator of LP as

$$\begin{pmatrix} \mathbf{c} \\ \mathbf{d} \\ \mathbf{y} \end{pmatrix} = \begin{pmatrix} H \\ I - GH \\ A \end{pmatrix} \mathbf{x}. \quad (4)$$

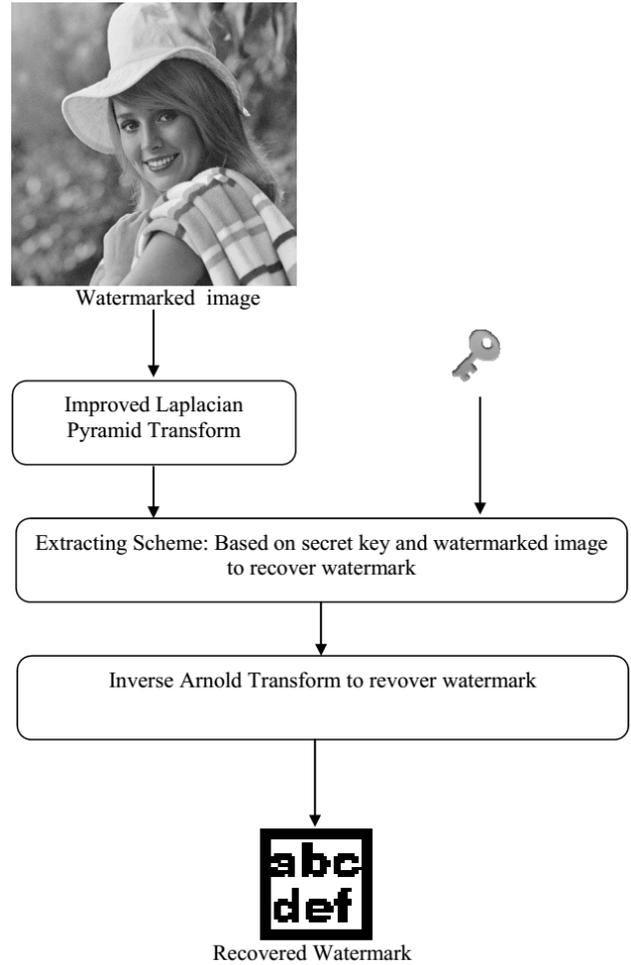


Figure 2: The proposed watermark extracting scheme.

The inverse transform of the LP is shown Figure 3(b) in which  $\hat{\mathbf{x}} = G\mathbf{c} + \mathbf{d}$  and, thus, one has

$$\hat{\mathbf{x}} = \begin{pmatrix} G & I \end{pmatrix} \begin{pmatrix} \mathbf{c} \\ \mathbf{d} \end{pmatrix}. \quad (5)$$

It has been shown in [15] that LP can be perfectly reconstructed with any pair of filters  $H$  and  $G$ .

### 2.1.2 Reconstruction using projection

A new reconstruction method is shown in Figure 4 [15]. From Figure 4, the improved inverse transform of LP can be written as

$$\hat{\mathbf{x}} = \begin{pmatrix} G & I - GH \end{pmatrix} \begin{pmatrix} \mathbf{c} \\ \mathbf{d} \end{pmatrix}. \quad (6)$$

Let  $S_2 = \begin{pmatrix} G & I - GH \end{pmatrix}$  be a transform matrix for the reconstruction algorithm. From Equations (4) and (6), we have  $S_2A = I - GH + (GH)^2$ . Thus,  $S_2$  is a left inverse of  $A$  if and only if  $GH = (GH)^2$ , i.e.,  $GH$  is a projector. The projection condition is

$$HG = I \quad (7)$$

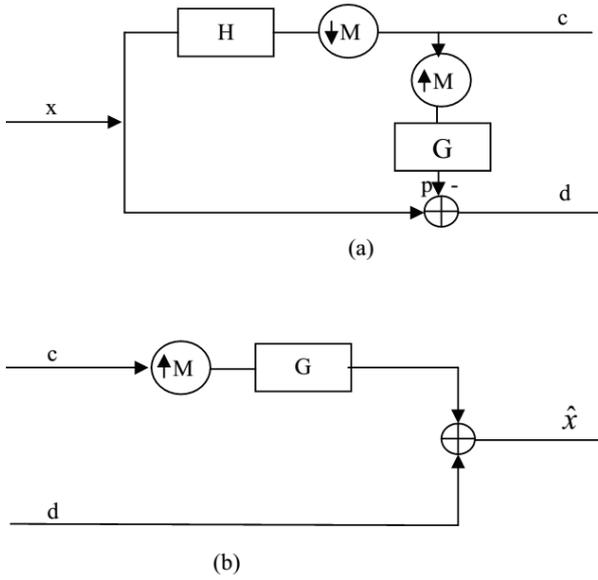


Figure 3: The typical Laplacian pyramid transform: (a) Analysis scheme; (b) Synthesis diagram.

or

$$\langle \tilde{h}[\cdot - Mk], g[\cdot - Ml] \rangle = \delta[k - l] \forall k, l \in Z^d. \quad (8)$$

Any filters  $H$  and  $G$  are called bi-orthogonal filters if they satisfy condition (8). The reconstruction scheme in Figure 4 is equivalent to an inverse transform of the LP if and only if two filters  $H$  and  $G$  are bi-orthogonal with given sampling lattice  $M$ . That is, the prediction operator of the  $LP(GH)$  is a projector. In this paper, we use the  $9 - 7$  bi-orthogonal filters whose coefficients are shown in Table 1. It is important to evaluate the reconstruction performance of the two methods in Figure 3(b) (namely, REC-1) and Figure 4 (REC-2). Suppose that one wants to approximate  $\mathbf{x}$  given  $\hat{\mathbf{y}} = A\mathbf{x} + \boldsymbol{\eta}$ . Without information about the error  $\boldsymbol{\eta}$ ,  $\hat{\mathbf{x}}$  is chosen such that the residual  $\|A\hat{\mathbf{x}} - \hat{\mathbf{y}}\|$  is minimized. Using this measurement to evaluate the reconstruction performance, reference [15] showed that REC-2 outperforms REC-1.

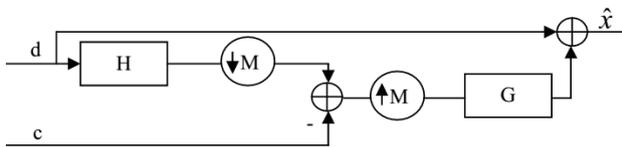


Figure 4: New reconstruction diagram for the LP scheme [14].

## 2.2 The Arnold transform

To provide a improved security for the watermark, the Arnold transform is adopted to make the watermark uncertain. With the Arnold transform, the watermark cannot be

defined even when it is detected. This transform also improves the robustness of the watermark. The Arnold transform function is given by [12]

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \text{ mod } N \quad (9)$$

where  $N$  is the watermark image size and the point  $(x', y')$  is a shifted version of point  $(x, y)$ .

## 2.3 The watermark embedding scheme

To embed the watermark into the LP transform domain, the improved LP decomposes the host image into multi-scale images. Since the Human Vision System (HVS) is very sensible to the low frequency coefficients, the watermark should be embedded into the high frequency coefficients in order to increase invisibility. However, the common image processing attacks normally affect the high frequencies of the image signals. Thus, robustness is improved if the watermark is inserted into the low frequencies. It is worth noting that robustness plays an important role for applications of protecting digital image copyrights. Thus, to increase the embedded watermark size and to enhance the robustness of the proposed algorithm, we have investigated the embedding of watermark into the prediction error coefficients  $\mathbf{d}$  at both low and middle frequencies from level number 5 ( $\mathbf{d}_5$ ) (from low frequency to high frequency: 5, 4, 3, 2, 1) as shown in Figure 5. The scheme for embedding watermark is described in detail in Algorithm 2 in which  $\ell$  is the decomposed level;  $\mathbf{d}_\ell$  is image of the prediction error at level  $\ell$ ;  $p_1$  and  $p_2$  are position parameters of  $\mathbf{d}_\ell$ ;  $k$  is position parameter of watermark. Function  $moveNext(d(\ell, p_1, p_2))$  returns the next coefficient of  $d(\ell, p_1, p_2)$ ,  $\beta$  is a metric of embedment strength,  $\max(\mathbf{d}_\ell)$  is the largest coefficient of prediction error of level  $\ell$ . Each bit of binary watermark is embedded in an improved LP coefficient. This coefficient is determined as follows. In each level ( $\ell$ ) selected to embed the watermark, we calculate a threshold value  $T_\ell = \beta \times \max(\mathbf{d}_\ell)$ . The value of  $T_\ell$  affects on the invisibility and robustness of watermarking schemes [11]. If the value of  $T_\ell$  is large, robustness will be strong, vice versa. The value of  $\beta$  belongs to  $0 < \beta \leq 1$ . Because the low frequency coefficients are used to embed the watermark in the proposed method, to balance between the invisibility and the robustness, the value of  $\beta$  is set to 0.2 [16] for all levels. The selected coefficient to embed a bit of watermark depends on the value of embedded bit and the value of examining coefficient as compared with  $T_\ell$ . If it is not the case where the coefficient do not satisfies the predefined conditions, the next coefficient will be considered. The positions of the selected coefficients are recorded in order to reuse in the watermark extracting scheme. The watermark embedding scheme is summarized in Algorithm 2.

Table 1: The 9 – 7 bi-orthogonal filters with coefficients.

$n$	0	$\pm 1$	$\pm 2$	$\pm 3$	$\pm 4$
$h[n]$	0.852699	0.377403	-0.110624	-0.023894	0.037828
$g[n]$	0.788486	0.418092	-0.040689	-0.064539	

### 2.4 An algorithm for extracting watermarks

To extract the watermarks, the watermarked image firstly is transformed into the improved LP domain. Based on the information recorded in Algorithm 2 about the positions selected to embed the binary watermark, we calculate and determine whether the bit at this position is bit 0 or bit 1. Second, as similar as in the scheme for embedding watermark, the threshold value is calculated by Equation (11). Third, we obtain sequently the positions recorded in the scheme for embedding watermark to seek the coefficient selected to embed a bit watermark. Finally, each coefficient will be processed and compared to the parameter  $T_{\ell}'$  to decide whether the bit embedded in this coefficient is bit 0 or bit 1. The description in detail of the extracting algorithm is represented in Algorithm 3 in which  $\ell$  is the decomposed level;  $\mathbf{d}_{\ell}'$  is image of the prediction error at level  $\ell$ ;  $p_1$  and  $p_2$  are position parameters of  $\mathbf{d}_{\ell}$ ;  $k$  is position parameter of watermark.

### 2.5 Performance metrics of an image watermarking algorithm

To measure the invisibility and robustness of the watermarking scheme, four parameters, namely the ratio of peak signal to noise (PSNR), the normalized correlation (NC), the structural similarity index (SSIM) and the execution time for embedding watermark and extracting watermark are typically considered [11, 12]. To assess the difference between the original image and the processed or attacked one, we can use three first metrics. Assume that the dimension of the images is  $M \times N$  and the pixels of the original image and of the watermarked images are  $X_{ij}$  and  $W_{ij}$ , respectively. To measure the invisibility between the original gray image and the watermarked one, we can use the PSNR defined by

$$PSNR = 10 \log_{10} \frac{(255)^2}{MSE} dB \tag{12}$$

where the mean square error (MSE) is given by

$$MSE = \frac{1}{M * N} \sum_{i=1}^M \sum_{j=1}^N (X_{ij} - W_{ij})^2 \tag{13}$$

On the other hand, the NC can measure the difference between original watermark and extracted watermark. Thus, to assess the robustness between the original watermark

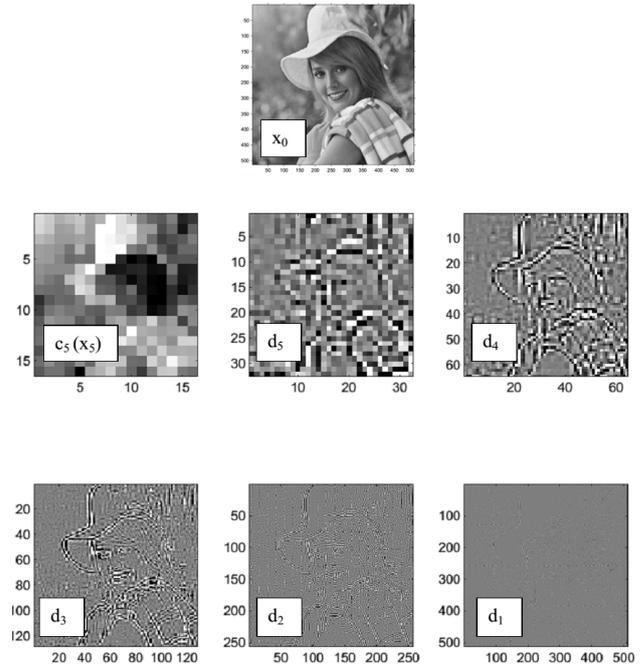


Figure 5: The results of decomposition host image by using the improved LP transform with 5 levels, and images of improved LP.

and recovered watermark, the NC can be used. The formula of NC is defined by

$$NC = \frac{\sum_{i=1}^M \sum_{j=1}^N W_{oij} W_{rij}}{\sqrt{\sum_{i=1}^M \sum_{j=1}^N W_{oij}^2} \sqrt{\sum_{i=1}^M \sum_{j=1}^N W_{rij}^2}} \tag{14}$$

where the pixels of the original watermark and the extracted watermark image of  $M \times N$  dimension are  $W_{oij}$  and  $W_{rij}$ , respectively. It is obvious that NC has the values from 0 to 1, and NC value of 1 reveals the best robustness. The robustness of watermarking schemes is also reflected by the NC when watermarked image is attacked on.

Structural Similarity index (SSIM) is commonly used to measure the similarity between two images [18]. Three components including luminance, contrast and structures are compared to compute SSIM. The value of SSIM is between 0 and 1, where 1 means two image identical and 0 means two image totally different. At each step, the local window is used to calculate the local statistics and SSIM index. Final local SSIM measure is the product of three

---

**Algorithm 2 :** The proposed scheme for embedding watermark

---

- 1: Analyze the host image by using the improved Laplacian pyramid transform toolbox [15] as shown in Figure 5 (decomposed level = 5). The binary watermark is embedded into the prediction error from level number 5 until the last bit of watermark is processed;
- 2: Scramble the watermark by the Arnold transform.
- 3: In each level ( $\ell = 5, 4, 3, 2, 1$ ), we calculate the threshold by using the following equation:

$$T_\ell = \beta \times \max(\mathbf{d}_\ell) \quad (10)$$

where  $\max(\mathbf{d}_\ell)$  is the largest Laplacian pyramid coefficients of level  $\ell$  and  $\beta$  is a strength parameter.

- 4: Set  $\ell = 5, k = 1, W(1)$  is the first bit of the binary watermark
  - 5: **while**  $\ell > 0$  and  $k \leq \text{size of } (W)$  **do**
  - 6:   **if**  $W(k) == 0$  **then**
  - 7:     **while** NOT  $(0 < d(\ell, p_1, p_2) < \frac{T_\ell}{2}$  or  $T_\ell < d(\ell, p_1, p_2) < \frac{3T_\ell}{2})$  **do**
  - 8:        $\text{moveNext}(d(\ell, p_1, p_2));$
  - 9:     **end while**
  - 10:     Set  $d(\ell, p_1, p_2) = d(\ell, p_1, p_2) - \text{mod}(d(\ell, p_1, p_2), T_\ell) + \frac{T_\ell}{4};$
  - 11:   **else**
  - 12:     **while** NOT  $(\frac{T_\ell}{2} < d(\ell, p_1, p_2) < T_\ell)$  **do**
  - 13:        $\text{moveNext}(d(\ell, p_1, p_2));$
  - 14:     **end while**
  - 15:     Set  $d(\ell, p_1, p_2) = d(\ell, p_1, p_2) - \text{mod}(d(\ell, p_1, p_2), T_\ell) + \frac{3T_\ell}{4};$
  - 16:   **end if**
  - 17:   Record this position  $(\ell, p_1, p_2)$  in array  $IND$
  - 18:    $k = k + 1$
  - 19: **end while**
  - 20: Use the inverse Laplacian pyramid to reconstruct the watermarked image shown in Figure 4.
  - 21: **Output:** watermarked image.
- 

components: luminance, contrast and structure as follows:

$$SSIM = \frac{(2 \cdot \gamma_x \cdot \gamma_y + C_1)(2 \cdot \rho_{xy} + C_2)}{(\gamma_x^2 + \gamma_y^2 + C_1)(\rho_x^2 + \rho_y^2 + C_2)} \quad (15)$$

where:

- $\gamma_x$  and  $\gamma_y$  are weighted means from original and degraded image,
- $\rho_x$  and  $\rho_y$  are weighted variances from original and degraded image.
- $\rho_{xy}$  is similarly defined as weighted covariance between original and degraded image.
- $C_1$  and  $C_2$  are constants.

Another metric used to evaluate our proposed method is execution time which is the amount of time required to embed a watermark into the host image, and then extract it.

---

**Algorithm 3 :** The proposed scheme for extracting watermark

---

- 1: Apply the improved Laplacian pyramid transform on watermarked image to get improved LP coefficient  $\mathbf{d}'$ .
- 2: Determine the threshold

$$T'_\ell = \beta \times \max(\mathbf{d}'_\ell) \quad (11)$$

where  $\max(\mathbf{d}'_\ell)$  is the largest Laplacian pyramid coefficients of level  $\ell$  and  $\beta$  is a strength parameter.

- 3: Set  $k = 1, IND$  is array used to record  $(\ell, p_1, p_2)$  in the embedding scheme,  $W'$  is a extracted watermark variable.
  - 4: **while**  $k \leq \text{size of } (W)$  **do**
  - 5:    $(\ell, p_1, p_2) = IND(k)$
  - 6:   **if**  $\text{mod}(d'(\ell, p_1, p_2), T'_\ell) \geq \frac{T'_\ell}{2}$  **then**
  - 7:      $W'(p) = 1$
  - 8:   **else**
  - 9:      $W'(p) = 0$
  - 10:   **end if**
  - 11:    $k = k + 1$
  - 12: **end while**
  - 13: Do the inverse Arnold transform to  $W'$  to get the recovered watermark.
  - 14: **Output:** extracted watermark.
- 

### 3 Experiment results

We carried out three experiments to evaluate the performance of the proposed watermarking scheme. In the experiments, we use the MATLAB R2013a as experimental platform on an Intel Core i5 – 2450M CPU@2.50GHz personal computer with 4 GB RAM. The method in [17] has been improved in this paper by using combination the low frequency and the middle frequency to embed the watermark ( $d_5, d_4$  and  $d_3$  as shown in Figure 5 are used). In addition, the watermark also has the size larger than that one in [17]. The host image is transformed into the frequency domain and is reconstructed to its spatial domain by using the improved Laplacian pyramid. As shown in [17, 19], the 9 – 7 bi-orthogonal filters with five levels of pyramidal decomposition are used to decompose the host image. The value for  $\beta$  is set to 0.2 as similar to those in [17]. The binary watermark after applying the Arnold function to enhance the security is embedded into the prediction errors of transformed domain.

#### 3.1 Experiment under JPEG lossy compression and Gaussian low pass filtering attacks

In this experiment, the  $512 \times 512$  pixels Lena gray image in Figure 6 (a) is used as a host image. Figure 6 (b) is used as a binary watermark whose size is  $32 \times 32$  pixels. Figure 6 (c) and Figure 6 (d) are the watermarked image and recovered watermark without any attacks, respectively.

Although in the proposed scheme, the low frequency sub-bands is exploited to embed the watermark, the PSNR between the host image and the watermarked image is quite high (PSNR = 47.016 dB as presented in Figure 6 (c)). In addition, it is very difficult to distinguish by human vision between the host image and the watermarked image. Furthermore, the proposed scheme can extract the watermark perfectly (NC=1) in the condition of no attacks.

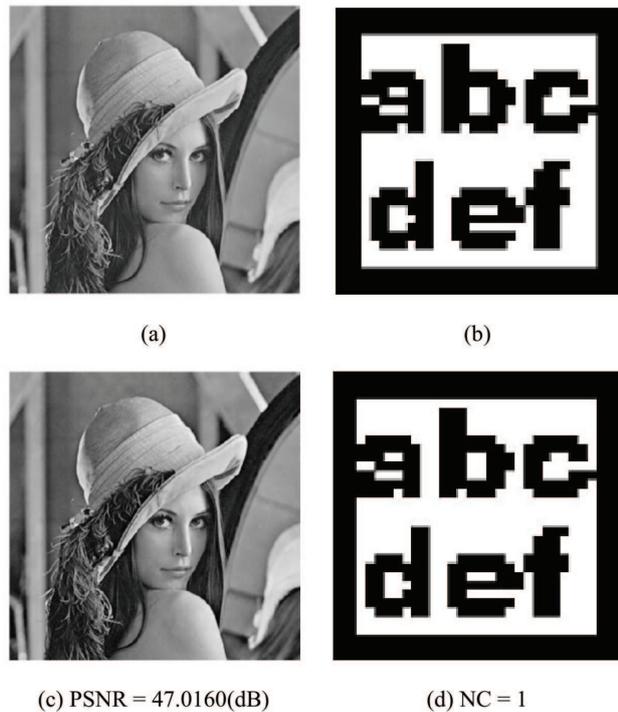


Figure 6: (a) Lena host image, (b) original binary watermark, (c) watermarked image, (d) recovered watermark.

The robustness of the proposed scheme was tested by JPEG lossy compression and Gaussian low pass filtering. The experimental results are compared with those in [11, 12] as shown in Table 2, Table 3, Figure 7 and Figure 8. The fidelity of the watermarked image after being attacked is evaluated by using the PSNR while the quality of the extracted watermark for some attacks is evaluated by the NC. The experimental results show that in terms of the robustness, the proposed scheme is superior to those in [11] under both JPEG lossy compression and Gaussian low pass filtering attacks. The robustness comparison between our proposed method with those in [12] is shown in Table 2. It can be seen from Table 2 that the NCs of the proposed scheme are higher than those by the method in [12] for almost all of the cases. Observing from Figure 7 and Figure 8, we found that the quality of the extracted watermarks under JPEG lossy compression attacks and Gaussian low pass filtering attacks are very high. Figure 7 shows that the quality of the extracted watermarks is reduced as the compression ratio increases. As shown in Table 2, the PSNR of the watermarked image in the condition of no attack of the proposed scheme is comparable to those in [11, 12] and the

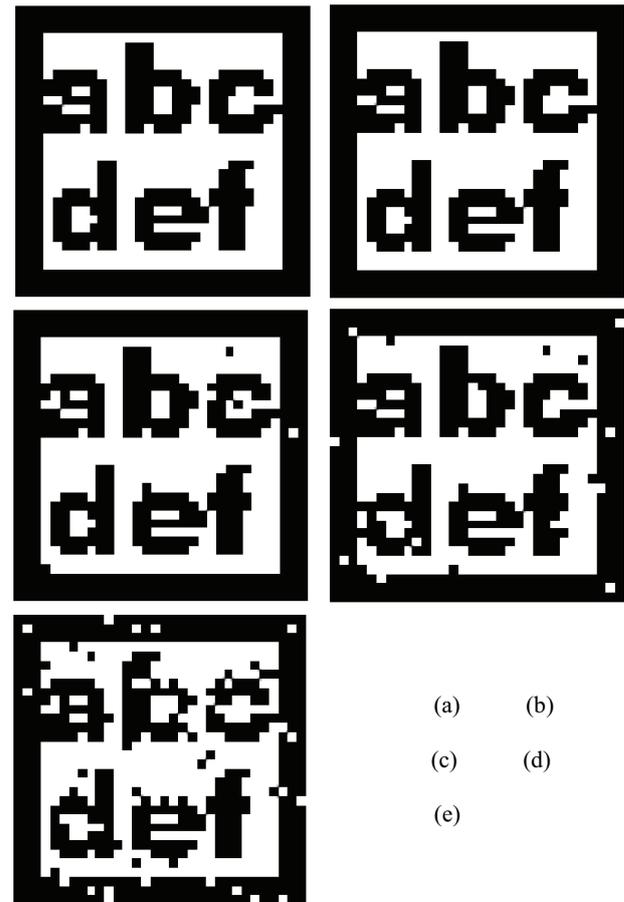


Figure 7: The extracted watermarks under JPEG lossy compression attacks: (a) no attack, (b)  $Q = 50$ , (c)  $Q = 30$ , (d)  $Q = 20$ , (e)  $Q = 15$

quality of the watermarked image is acceptable. It is noted the methods in [11, 12] used the FDCT as the key transform in their watermarking schemes. By simulation, we also showed that the computational time of decomposition and reconstruction of our improved LP is lower than those of FDCT as listed in Table 5. This illustrates the computational efficiency of our proposed method.

### 3.2 Experiment under intentional attacks

In this experiment, we evaluate the performance of the proposed method in case of being attacked by intentional cutting 25% of the watermarked images. In cutting attacks, the attached pixels of the images are changed to black. Three images, namely, Baboon in Figure 9(a), Peppers in Figure 9(d) and Boat in Figure 9(g), are used as host images in this test. The binary watermark is the same as the watermark in Experiment 3.1 (i.e., Figure 6(b)). The PSNR is employed to measure the quality of watermarked image after being cut 25%; The PSNR, NC and SSIM are employed to evaluate the robustness of the proposed method after being cutting intentional attacks. The results of this test are depicted in Figure 9 and are listed in Table 4. Although the water-

Table 2: The comparison of the robustness against JPEG lossy compression of the proposed scheme with those in [11, 12].

Attacks	Reference [11]		Reference [12]		Proposed scheme	
	PSNR(dB)	NC	PSNR(dB)	NC	PSNR(dB)	NC
Noattack	50.12	1	60.80	1	47.02	1
Q=50	36.33	0.998	24.97	0.991	35.31	1
Q=30	35.17	0.995	24.89	0.974	34.03	0.991
Q=20	33.51	0.971	24.67	0.947	32.47	0.972
Q=15	32.30	0.945	24.48	0.911	31.81	0.933

Table 3: The comparison of the robustness against Gaussian low pass filtering of the improved scheme with those in [11].

$\sigma(window)$	Proposed scheme			Reference [11]	
	Recovered watermark	PSNR(dB)	NC	PSNR(dB)	NC
0.5(3)	Figure 8(a)	39.6934	0.9977	40.8289	0.9914
1.5(3)	Figure 8(b)	32.0616	0.9748	32.4277	0.9704
0.5(5)	Figure 8(c)	39.7431	0.9954	40.8027	0.9914
1.5(5)	Figure 8(d)	29.4963	0.9537	29.9117	0.9113
5.0(3)	Figure 8(e)	31.5318	0.9725	28.7562	0.86582

marked images are significantly destroyed by the cutting attacks which reveals by the low PSNR of the watermarked image, the watermark still can be extracted with the acceptable NC and high SSIM. In addition, the extracted watermark can be recognized by human vision. This result verifies the robustness of the proposed algorithm.

### 3.3 Experiment for measuring the computational time

This experiment evaluates the computational time of the proposed method using an improved LP transform (5 levels) as compared to other methods using FDCT [11, 12]. The host images are gray images of  $512 \times 512$  pixels, including the Lena image in Figure 6 (a), the Elaine image in Figure 5 ( $x_0$ ) and the Peppers image in Figure 9 (d). The processing time of three methods is listed in Table 4. On average, the processing time of our improved LP is about 0.0572 seconds for both decomposition and reconstruction, and it is significantly lower those with methods with using FDCT.

## 4 Conclusion

In this work, we have presented an improved watermarking scheme using the LP coefficients in low and middle frequency sub-bands to embed binary watermark. The results of this research showed that the performance of the proposed algorithm in terms of invisibility and robustness is better than those using 2D DWT and FDCT under JPEG lossy compression, Gaussian low pass filters, intentional cutting attacks. In addition, the proposed schemes require

less computational time of embedding and extracting watermark than the other. In addition, the proposed method is blind watermarking and high security and, thus, the watermark is only deducted by the legal users.

## References

- [1] X. L. Liu, C. C. Lin, and S. M. Yuan, "Blind dual watermarking for color images' authentication and copyright protection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 5, pp. 1047–1055, 2018. <https://doi.org/10.1109/tcsvt.2016.2633878>.
- [2] S. Khalighi, P. Tirdad, and H. Rabiee, "A contourlet-based image watermarking scheme with high resistance to removal and geometrical attacks," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 21, pp. 1–13, 2010. <http://dx.doi.org/10.1155/2010/540723>.
- [3] B. L. Gunjal and R. R. Manthalkar, "An overview of transform domain robust digital image watermarking algorithms," in *Journal of Emerging trends in Computing and Information Science*, 2010, pp. 37–42. [http://www.cisjournal.org/Download\\_pdf\\_2\\_5.aspx](http://www.cisjournal.org/Download_pdf_2_5.aspx).
- [4] J. Liu, Y. Xu, S. Wang, and C. Zhu, "Complex wavelet-domain image watermarking algorithm using L1-Norm function-based quantization," *Circuits, Systems, and Signal Processing*, pp. 1–19, 2017. <https://doi.org/10.1007/s00034-017-0607-5>.

Table 4: The empirical results of invisibility and robustness of proposed against 25% cutting attacks.

Host image	Attacked image	Recovered watermark		
	PSNR( <i>dB</i> )	NC	PSNR( <i>dB</i> )	SSIM
Baboon	8.7030	0.7092	6.1760	0.9926
Peppers	9.5256	0.7555	7.2700	0.9912
Boat	10.5776	0.8123	8.1164	0.9953
Average	9.6021	0.7590	7.1875	0.9930

Table 5: Comparison the processing times of 2 transforms: FDCT(wrapping) and Improved LP.

Host image	FDCT(wrapping)		Improved LP	
	Decomposition(s)	Reconstruction(s)	Decomposition(s)	Reconstruction(s)
Lena	0.4212	0.6864	0.0468	0.0312
Elain	0.4836	0.5304	0.0468	0.0780
Peppers	0.3744	0.5460	0.0780	0.0624
Average	0.4264	0.5876	0.0572	0.0572

- [5] L.-Y. Hsu and H.-T. Hu, "Robust blind image watermarking using crisscross inter-block prediction in the DCT domain," *Journal of Visual Communication and Image Representation*, vol. 46, no. Supplement C, pp. 33 – 47, 2017. <https://doi.org/10.1016/j.jvcir.2017.03.009>.
- [6] V. M. Potdar, S. Han, and E. Chang, "A survey of digital image watermarking techniques," in *INDIN '05. 2005 3rd IEEE International Conference on Industrial Informatics, 2005.*, 2005, pp. 709–716. <https://doi.org/10.1109/indin.2005.1560462>.
- [7] P. Tao and A. M. Eskicioglu, "A robust multiple watermarking scheme in the discrete wavelet transform domain," in *Internet Multimedia Management Systems V*, vol. 5601, 2004, pp. 133–144. <http://dx.doi.org/10.1117/12.569641>.
- [8] S. P. Singh, P. Rawat, and S. Agrawal, "A robust watermarking approach using DCT-DWT," *International Journal of Emerging Technology and Advanced Engineering*, vol. 2, pp. 300–305, 2012. [https://ijetae.com/files/Volume2Issue8/IJETAE\\_0812\\_52.pdf](https://ijetae.com/files/Volume2Issue8/IJETAE_0812_52.pdf).
- [9] N. S. Narawade and R. D. Kanphade Taiyue, "DCT based robust reversible watermarking for geometric attack," *International Journal of Emerging Trends and Technology in Computer Science (IJETTCS)*, vol. 1, pp. 27–32, 2012. <https://www.ijettcs.org/Volume1Issue2/IJETTCS-2012-07-25-025.pdf>.
- [10] W. Bender, W. Butera, D. Gruhl, R. Hwang, F. J. Paiz, and S. Pogreb, "Applications for data hiding," *IBM Systems Journal*, vol. 39, no. 3.4, pp. 547–568, 2000. <https://doi.org/10.1147/sj.393.0547>.
- [11] Z. y. Zhang, W. Huang, J. l. Zhang, H. y. Yu, and Y. j. Lu, "Digital image watermark algorithm in the curvelet domain," in *2006 International Conference on Intelligent Information Hiding and Multimedia*, 2006, pp. 105–108. <https://doi.org/10.1109/iuh-msp.2006.264965>.
- [12] H. P. J. Xu and J. Zhao, "Digital image watermarking algorithm based on fast curvelet transform," *Journal of Software Engineering and Applications*, vol. 3, no. 10, pp. 939–943, 2010. <https://doi.org/10.4236/jsea.2010.310111>.
- [13] S. C. Nguyen, K. H. Ha, and H. M. Nguyen, "An improved image watermarking scheme using selective curvelet scales," in *2015 International Conference on Advanced Technologies for Communications (ATC)*, 2015, pp. 445–450. <https://doi.org/10.1109/atc.2015.7388369>.
- [14] P. Burt and E. Adelson, "The laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, pp. 532–540, 1983. <https://doi.org/10.1109/tcom.1983.1095851>.
- [15] M. N. Do and M. Vetterli, "Framing pyramids," *IEEE Transactions on Signal Processing*, vol. 51, no. 9, pp. 2329–2342, 2003. <https://doi.org/10.1109/tsp.2003.815389>.
- [16] S. C. Nguyen, H. H. Kha, and H. M. Nguyen, "A new image watermarking scheme using contourlet transforms," in *The 3rd International Conference on Information Technology, Computer, And Electrical Engineering ( ICITACEE 2016 )*, 2016, pp. 283–288. <https://doi.org/10.1109/icitacee.2016.7892456>.

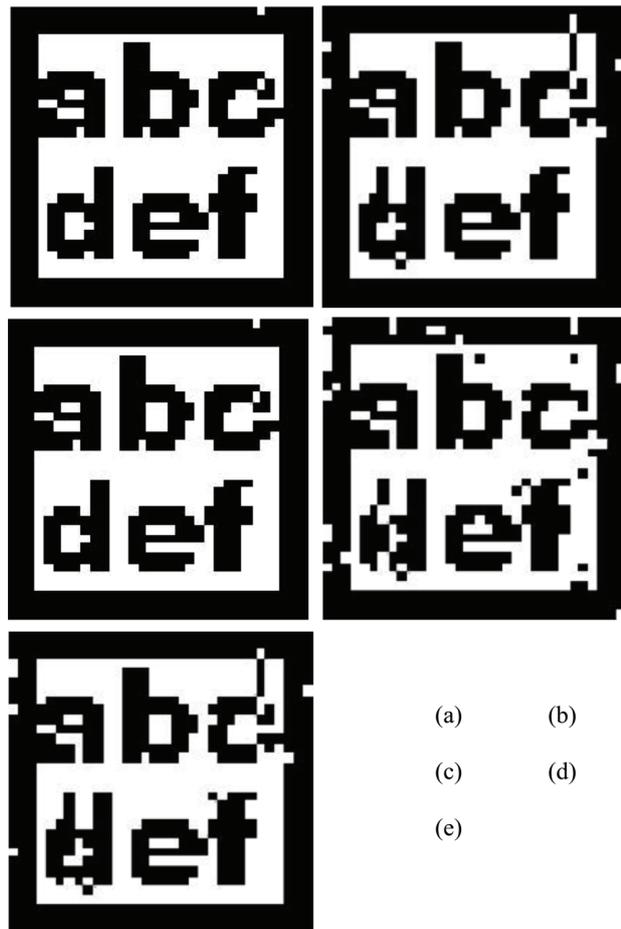


Figure 8: The extracted watermarks under the Gaussian low pass filtering: (a)  $\sigma(\text{window}) = 0.5(3)$ , (b)  $\sigma(\text{window}) = 1.5(3)$ , (c)  $\sigma(\text{window}) = 0.5(5)$ , (d)  $\sigma(\text{window}) = 1.5(5)$ , (e)  $\sigma(\text{window}) = 5.0(3)$ .

- [17] S. C. Nguyen, H. H. Kha, and H. M. Nguyen, “An efficient image watermarking scheme using the laplacian pyramid based on projection,” in *2017 International Conference on Recent Advances in Signal Processing, Telecommunications Computing (SigTelCom)*, 2017, pp. 103–108. <https://doi.org/10.1109/sigtelcom.2017.7849804>.
- [18] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. <https://doi.org/10.1109/tip.2003.819861>.
- [19] D. D. Y. Po and M. N. Do, “Directional multiscale modeling of images using the contourlet transform,” *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1610–1620, 2006. <https://doi.org/10.1109/ssp.2003.1289394>.

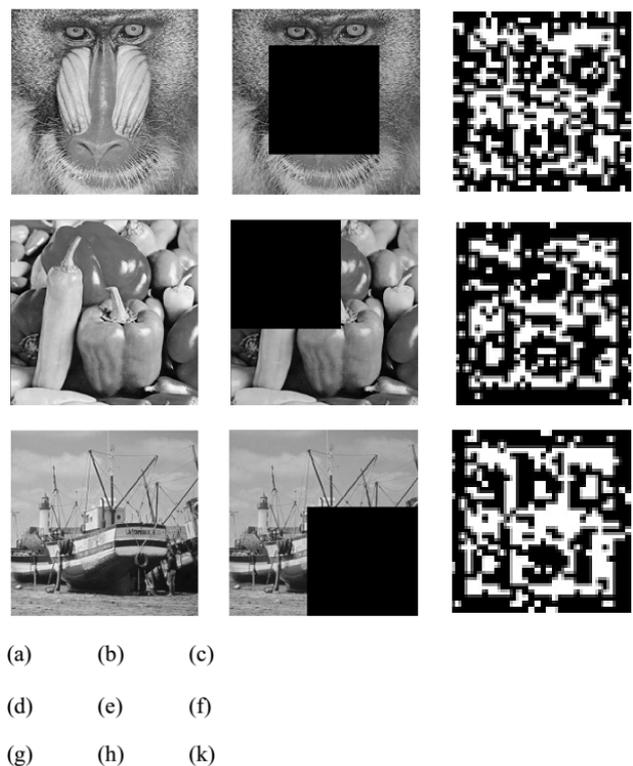


Figure 9: cropping 25% attacks, (a) The Baboon host image, (d) The Peppers host image, (g) The Boat host image; (b) The Baboon watermarked image, (e) The Peppers watermarked image, (h) The Boat watermarked image; (c) recovered watermark from the Baboon watermarked image, (f) recovered watermark from the Peppers watermarked image, (k) recovered watermark from the Boat watermarked image.

# Feature Level Fusion of Face and Voice Biometrics Systems Using Artificial Neural Network for Personal Recognition

Cherifi Dalila, El Affifi Omar Badis and Boushaba Saddek  
Institute of Electrical and Electronic Engineering, University of Boumerdes, Algeria  
E-mail: da.cherifi@univ-boumerdes.dz

Nait-Ali Amine  
Laboratoire Images, Signaux et Systèmes Intelligents (LISSI), Université Paris-EST, Vitry sur Seine, 94400, France  
E-mail: naitali@u-pec.fr

**Keywords:** bioinformatiscs, face, voice, multibiometric recognition system, fusion at feature level, Artificial Neural Network (ANN).

**Received:** November 23, 2018

*Lately, human recognition and identification has acquired much more attention than it had before, due to the fact that computer science nowadays is offering lots of alternatives to solve this problem, aiming to achieve the best security levels. One way is to fuse different modalities as face, voice, fingerprint and other biometric identifiers. The topics of computer vision and machine learning have recently become the state-of-the-art techniques when it comes to solving problems that involve huge amounts of data. One emerging concept is Artificial Neural networks. In this work, we have used both human face and voice to design a Multibiometric recognition system, the fusion is done at the feature level with three different schemes namely, concatenation of pre-normalized features, merging normalized features and multiplication of features extracted from faces and voices. The classification is performed by the means of an Artificial Neural Network. The system performances are to be assessed and compared with the K-nearest-neighbor classifier as well as recent studies done on the subject. An analysis of the results is carried out on the basis Recognition Rates and Equal Error Rates.*

*Povzetek: Z nevronske mrežo so kombinirani obraz in glas za biometrično identifikacijo.*

## 1 Introduction

Based on the fact that any biometric system has some weaknesses, it is difficult to obtain a system that accomplishes the four most desirable points for a biometric-based security system which are, Universality, Distinctiveness, Permanence and Collectability [1]. One way to overcome the limitations is through a combination of different biometric systems to reduce the classification problem which deals with the intra-class and inter-class variety [2]. Combinations of biometric traits are mainly preferred due to their lower error rates. Using multiple biometric modalities has been shown to decrease error rates, by providing additional useful information to the classifier. Fusion of these behavioral or physiological traits can occur in various levels. Different features can be used by a single system or separate systems which can operate independently and their decisions may be combined [3-6].

In this article, we have choiced Face and Voice as our biometric traits for several reasons, mainly because of their availability where people can get along with easily, regardless of gender and age. Also, because the data can be acquired simultaneously just by using a camera with an embedded microphone, this way, we avoid steps in data gathering like in the case of face and fingerprint or face and hand geometry, where the recognition algorithm might become time consuming and disables the real time functionality.

Many researchers have presented different multimodal biometric schemes for person verification using voice and face by using different fusion technique and data bases, the authors proposed different methods to extract the features from the face (Discrete Cosine Transform, grid-based lip motion, contour based lip motion, Morphological Dynamic, Link Architecture, 2D LDA, Eigenfaces, PCA, LDA and Gabor filter), and for the voice (Mel Frequency Cepstral Coefficients, Weighted Linear Prediction Cepstral Coefficients, Linear Prediction Coefficients and Linear Prediction Cepstral Coefficients) [7-14].

In this work the fusion is done at the feature level with three different schemes namely, concatenation, merging and multiplication of features extracted from faces and voices. The classification is performed by using two classifiers which are mainly K-Nearest-Neighbor and Artificial Neural Network. The first one is a classical classification method based on distance calculations, whereas the other is an intelligent system that learns in a way similar to the human brain. The complexity of the Neural Network gives it a flexibility and a capability to be tuned to better fit any type of data. In our work, we make a comparative study between the two stated classifiers to conclude whether ANN can be exploited to design better recognition systems.

The rest of the paper is structured as follow: In section 2, we dealt with feature extraction methods for

face and voice used in this work. In section 3, we presented our proposed fusion method at feature level based on Artificial Neural Networks and K-nearest-neighbor classifiers. In section 4, the experimental part is described, the results are provided and discussed. Finally, a conclusion of this work is highlighted in section 4.

## 2 Feature extraction

### 2.1 Face feature extraction

Face recognition is one of the few biometric methods that possess the merits of both high accuracy and low intrusiveness. It has the accuracy of a physiological approach without being intrusive. For this reason, it has drawn the attention of researchers in fields from security, psychology, and image processing, to computer vision [15]. Numerous algorithms have been proposed and developed for the purpose of Face recognition. These algorithms can be classified into three categories: Global-Appearance-based methods, Local-feature-based methods and Hybrid methods. There are methods that use the whole image of the face as a raw input to the learning process, others require the use of specific regions located on a face such as eyes, nose and mouth. There exist also methods that simply partition the input face image into blocks without considering any specific regions. In this work we mainly are going to use PCA and DCT [16].

- Principal Component Analysis (PCA) Method was developed by Turk and Pentland, it's a well-known face recognition method, known as eigenfaces, which drastically reduces the dimensionality of the original space and face detection and identification are carried out in the reduce space [17-19].
- Discrete Cosine Transform (DCT) Method is an invertible linear transform that can express a finite sequence of data points in terms of a sum of cosine functions oscillating at different frequencies [20-21]. Face recognition using DCT is divided into two stages training and classification. In the training stage, the face images are analyzed on block by block basis. The DCT coefficients with large magnitude are mainly located in the upper-left corner of the DCT matrix. Accordingly, we scan the DCT coefficient matrix in a zig-zag manner starting from the upper-left corner and subsequently convert it to a one-dimensional (1-D) vector [22].

### 2.2 Speech feature extraction

The speech signal conveys many levels of information to the listener (figure 1). At the primary level, speech conveys a message via words. But at other levels speech conveys information about the language being spoken and the emotion, gender and, generally, the identity of the speaker [23]. The general area of speaker recognition encompasses two more fundamental tasks.

**Speaker identification** is the task of determining who is talking from a set of known voices or speakers. The unknown person makes no identity claim and so the system must perform a 1:N classification. Generally, it is assumed the unknown voice must come from a fixed set

of known speakers, thus the task is often referred to as *closed-set* identification.

**Speaker verification** (also known as speaker authentication or detection) is the task of determining whether a person is who he/she claims to be (a yes/no decision). Since it is generally assumed that imposters (those falsely claiming to be a valid user) are not known to the system, this is referred to as an *open-set* task [23]. Depending on the level of user cooperation and control in an application, the speech used for these tasks can be either *text-dependent* or *text-independent*. In a text-dependent application, the recognition system has prior knowledge of the text to be spoken and it is expected that the user will cooperatively speak this text. In the other hand, in a text-independent application, there is no prior knowledge by the system of the text to be spoken, such as when using extemporaneous speech. Text-independent recognition is more difficult but also more flexible [23], this approach is considered in our work. It is inconvenient to use the whole speech directly as an input for biometric recognition systems. We instead use the features which represent the unique distinctive characteristics that make the difference between speakers for the following reasons [24]:

- The feature extraction process transforms the raw signal into feature vectors in which speaker-specific properties are emphasized and statistical redundancies are suppressed.
- With features extracted, we can avoid the problem of the curse of dimensionality.
- The signal during training and testing session can be greatly different due to many factors such as people voice change with time, health condition (e.g. the speaker has a cold), speaking rate and also acoustical noise and variation recording environment via microphone.

There is several feature extraction approaches for speech, the most popular are: Linear Predictive Analysis (LPC), Linear Predictive Cepstral Coefficients (LPCC), Perceptual Linear Predictive Coefficients (PLP), Relative Spectra filtering of log domain (RASTA), Mel-Frequency Cepstral Coefficients (MFCC).

#### 2.2.1 Mel-frequency cepstral coefficients

The MFCC feature extraction technique is the most popular approach used in speaker recognition systems today, it has been utilized intensively in literature [25-26] and others. The Mel scale was developed by Stevens and Volkman in 1940 as a result of a study of the human auditory perception. This method is capable of capturing phonetically important characteristics of the speech. MFCC are based on the well-known variation of the human ear's critical bandwidths with frequency. Steps of the MFCC extraction process are summarized in figure 2 [27].

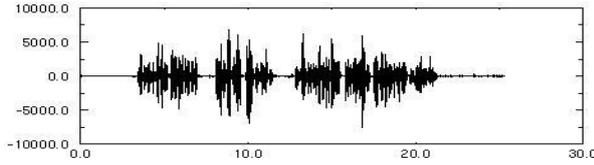


Figure 1: A sample of input speech signal[27].

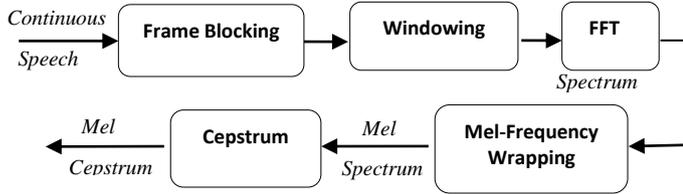


Figure 2: Block diagram of the MFCC process [27].

### 2.2.2 Vector Quantization (VQ)

Several state-of-the-art feature characterization and matching techniques have been developed and proposed in literature for speaker recognition. Dynamic Time Warping (DTW), Hidden Markov Modeling (HMM), and Vector Quantization (VQ). The last one was used in our project, because it is easy to implement. Vector Quantization (VQ) is a process of mapping vectors from a large vector space to some regions in that space. Each region is called a cluster that can be represented by its center which called a codeword. The set of all code words is called a codebook [27]. A speaker-specific VQ codebook is generated from any speaker by clustering his/her training acoustic vectors. The distance from a vector to the closest codeword of a codebook is called VQ-distortion.

### 2.2.3 Feature scaling

Since the face features vary in a scale of [0, 255], and voice features in a complete different scale [-10,14], a feature normalization must be performed to map the values from their ranges to a range of [0,1] in order to prevent one modality from contributing more than the other in the learning process. We have used the Min-Max normalization rule.

## 3 Data fusion at feature level

### 3.1 Proposed fusion schemes

In feature level fusion, feature sets originating from multiple information sensors are integrated into a new feature set. For non-homogeneous compatible feature sets, such as features of different modalities like face and speech as is presented in this article, a single feature vector can be obtained by concatenation [1, 20]. The new feature vector now has a higher dimensionality which increases the computational load. It is reported that a significantly more complex classifier design might be needed to operate on the concatenated data set at the feature level space.

The fusion at the feature level is expected to perform better in comparison with the fusion at the score level and decision level. The main reason is that the feature level contains richer information about the raw biometric data [28]. It is to be noted that a normalization may be necessary because of the **non-homogeneity** of the different traits used in the Multibiometric system. In the present work, we consider performing a data fusion at the feature level between face and voice. This is to be done in three different ways.

#### 3.1.1 Fusion by concatenation (pre-normalized features)

In this Fusion, we concatenate features of a Face sample ( $F_{ij}$ ) with features of a Voice sample ( $V_{ij}$ ) to get one large sample, without normalization of the features, taking  $m$  samples with  $n$  features of each.

$$\begin{bmatrix} F_{11} & F_{12} & \dots & F_{1n} \\ F_{21} & F_{22} & \dots & F_{2n} \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ F_{m1} & F_{m2} & \dots & F_{mn} \end{bmatrix} \text{Concatinated with} \begin{bmatrix} V_{11} & V_{12} & \dots & V_{1n} \\ V_{21} & V_{22} & \dots & V_{2n} \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ V_{m1} & V_{m2} & \dots & V_{mn} \end{bmatrix}$$

$$\text{Giving} \Rightarrow \begin{bmatrix} F_{11} & F_{12} & \dots & F_{1n} & V_{11} & V_{12} & \dots & V_{1n} \\ F_{21} & F_{22} & \dots & F_{2n} & V_{21} & V_{22} & \dots & V_{2n} \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots & \vdots & \vdots & \dots & \vdots \\ F_{m1} & F_{m2} & \dots & F_{mn} & V_{m1} & V_{m2} & \dots & V_{mn} \end{bmatrix}$$

Figure 3: Fusion by Concatenation.

This has been previously done and stated in literature [29], we apply it in order to see the impact of data normalization and its absence.

#### 3.1.2 Fusion by merging (normalized features)

This is to be done by alternatively placing one face feature, followed by one voice feature, until all features are placed one next to the other with normalized features.

$$\begin{bmatrix} F_{11} & F_{12} & \dots & F_{1n} \\ F_{21} & F_{22} & \dots & F_{2n} \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ F_{m1} & F_{m2} & \dots & F_{mn} \end{bmatrix} \text{Merged with} \begin{bmatrix} V_{11} & V_{12} & \dots & V_{1n} \\ V_{21} & V_{22} & \dots & V_{2n} \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ V_{m1} & V_{m2} & \dots & V_{mn} \end{bmatrix}$$

$$\text{Giving} \Rightarrow \begin{bmatrix} F_{11} & V_{11} & \dots & F_{1n} & V_{1n} \\ F_{21} & V_{21} & \dots & F_{2n} & V_{2n} \\ \vdots & \vdots & \dots & \vdots & \vdots \\ \vdots & \vdots & \dots & \vdots & \vdots \\ F_{m1} & V_{m1} & \dots & F_{mn} & V_{mn} \end{bmatrix}$$

Figure 4: Fusion by Merging.

### 3.1.3 Fusion by multiplication (normalized features)

This is to be done by multiplying pre-normalized Face features with pre-normalized Voice features element-wise. Then we normalize the resulting product matrix. We did not find a theoretical background for this fusion scheme except considering that features multiplication can be some sort of polynomial terms [30].

$$\begin{matrix}
 \begin{bmatrix} F_{11} & F_{12} & \dots & F_{1n} \\ F_{21} & F_{22} & \dots & F_{2n} \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ F_{m1} & F_{m2} & \dots & F_{mn} \end{bmatrix} & * & \begin{bmatrix} V_{11} & V_{12} & \dots & V_{1n} \\ V_{21} & V_{22} & \dots & V_{2n} \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ V_{m1} & V_{m2} & \dots & V_{mn} \end{bmatrix} \\
 \Leftrightarrow \text{Giving} & & \begin{bmatrix} F_{11} * V_{11} & F_{12} * V_{12} & \dots & F_{1n} * V_{1n} \\ F_{21} * V_{21} & F_{22} * V_{22} & \dots & F_{2n} * V_{2n} \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ F_{m1} * V_{m1} & F_{m2} * V_{m2} & \dots & F_{mn} * V_{mn} \end{bmatrix}
 \end{matrix}$$

Figure 5: Fusion by Multiplication.

Each of the resulting fused data will be fed to our designed Neural Network system for classification. The results will be compared with the performance of a K-NN classifier as shown in Table 1.

	Face	Voice	Fusion	Feature scale
Method 1	Raw Pixels	MFCC + VQ	Concatenation	Pre-Normalized
Method 2	Raw Pixels	MFCC + VQ	Merged	Normalized
Method 3	Raw Pixels	MFCC + VQ	Multiplied	Normalized
Method 4	PCA	MFCC + VQ	Concatenation	Normalized
Method 5	DCT	MFCC+VQ	Concatenation	Normalized

Table 1: Proposed fusion methods to be experimented.

## 3.2 Classifiers

### 3.2.1 K-nearest neighbor algorithm

The idea in k-Nearest Neighbor methods is to identify  $k$  samples in the training set whose independent variables  $x$  are similar to  $u$ , and to use these  $k$  samples to classify this new sample into a class,  $v$ . If all we are prepared to assume is that  $f$  is a smooth function, a reasonable idea is to look for samples in our training data that are near it (in terms of the independent variables) and then to compute  $v$  from the values of  $y$  for these samples. When we talk about neighbors, we are implying that there is a distance or dissimilarity measure that we can compute between samples based on the independent variables. One way to perform this task is to use the most popular measure of distance: Euclidean distance.

### 3.2.2 Artificial neural networks

Neural networks are algorithms that are patterned after the structure of the human brain. They contain a series of mathematical equations that are used to simulate biological processes such as learning and memorizing.

In a neural network, the goal as in all modeling techniques (such as Linear regression, Logistic regression, Survival analysis or time-series analysis ...), is predicting an outcome based on the values of some input variables stated that ANNs could be used as alternatives to the foregoing techniques. Neural networks can have one or multiple outputs. In this work, we are dealing with multi-class classification problem, where each person (Face and Voice) is a distinct class, hence the use of a multiple output Neural Network. Although many different types of neural network training algorithms have been developed, we preferred to stick with the famous “back-propagation” algorithm, which is the most popular used technique [31-34] and we have considered the Logistic activation function in our network design represented in Figure 6.

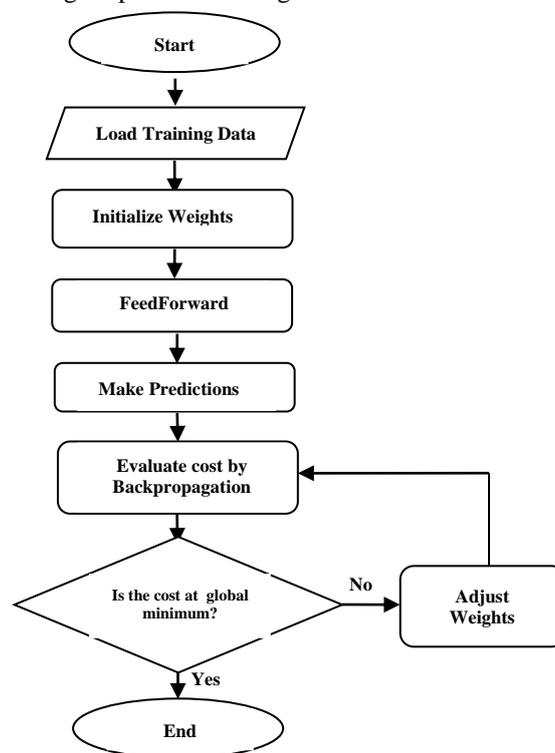


Figure 6: Neural network training.

## 4 Experimental & results

In the present work, the major aim is to realize a multibiometric system based on a fusion of two main modalities, Face and Voice. This is to be done on the feature level. An Artificial Neural Network is to be designed for the sake of classification. The performance of this system is then compared to the k-NN classification approach involving the use of some classical methods (PCA and DCT) for the face, and MFCC with Vector Quantization for voices.

A fusion is done at the feature level for each of the following systems (Table 1), and then fed to a k-NN classifier. We consider applying this approach on many databases, and compare performances with respect to the Artificial Neural Network design.

## 4.1 Database description

In this work, we have run into the problem of missing a database that contains both the face and voice of the same person, because it is unlikely for a subject to give away two or three of his identity modalities at once for the sake of a bare scientific experiment. This is generally justified by security and anonymity reasons. In order for us to approach this issue, we have followed some works in literature, in which the authors have combined two or three datasets. The first set is for one modality, taken from a group of subjects at some circumstances, the other set is for another modality taken from a dissimilar group of people at completely different circumstances. Then each modality from set 1 is assigned to the other modality from set 2, thus the fusion is performed by concatenation. The database formed by the procedure just described is usually referred to as a virtual database [25-26].

### 4.1.1 Face databases: ORL database (AT&T)

There are ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). A preview of the faces is in (Figure 7). The files are in PGM format. The size of each image is 112x92 pixels, with 256 grey levels per pixel.



Figure 7: Preview of the ORL database images.

### 4.1.2 Face databases: FEI database

The FEI face database is a Brazilian face database that contains a set of face images taken between June 2005 and March 2006. There are 14 images for each of 200 individuals, a total of 2800 images. All images are colorful and taken against a white homogenous background in an upright frontal position with profile rotation of up to about 180 degrees. Scale might vary about 10% and the original size of each image is 640x480x3 pixels. All faces are mainly represented by students and staff at FEI, between 19 and 40 years old with distinct appearance, hairstyle, and adorns. The number of male and female subjects are exactly the same and equal to 100. Figure 8 shows a sample of image variations from the FEI face database.



Figure 8: Preview of the FEI database images.

### 4.1.3 Speech database

We have collected samples that are 12 minutes long from different people reading books from the internet. The utterances were text-independent. Then we adjusted the sampling frequency of every sample to 11025 Hz using audio enhancement software (Audacity). After that, we cropped the long samples at a length of less than 14 seconds making 48 samples per person.

### 4.1.4 Neural network design

Since there is no rule of thumb for choosing the number of hidden layers as well as the number of neurons contained inside them, we tried a set of configurations with multiple numbers of layers and neurons, and analyzed the behavior of the networks designed at each time.

## 5 Experiments

In order to evaluate the performance of the proposed methods, we have used some standard indices for assessment. The false acceptance rate (FAR), is the measure of the likelihood that the biometric security system will incorrectly accept an access attempt by an unauthorized user. The false rejection rate (FRR) is the measure of the likelihood that the biometric security system will incorrectly reject an access attempt by an authorized user. The Equal Error Rate (EER) is defined as the point where the value of FAR equals the value of FRR in the Receiver Operating Curve (ROC) which plots FAR versus FRR.

### 5.1 Experiment I: (ORL without external effects + voice)

We have downsized the ORL images to 40x40 pixels, in order to minimize the amount of calculations as compared to 112x92. The number of subjects is 40. We used ORL images without any external effects, samples of speech are assigned to those face samples for each subject. A detailed description of this database used for this experiment, containing the matrices dimensions before and after fusion (Table 2). The results of this experiment are shown in (Table 3) describing the recognitions rates and equal error rates.

In terms of recognition rate, when trained and tested without external effects, ANN gave better results than K-NN with an average of (96.33 vs 92.66%) still with an insignificant difference ( $p=0.081 > 0.05$ ). The proposed method 2 (Raw Faces & MFCC + VQ) merged and normalized hit the best accuracy (99.16%). This is because the configuration of the network enables it to fit well trained data and generalize to the test data. In terms of equal error rate, method 5 (DCT of Faces & MFCC + VQ) on K-NN outperformed all the methods (1.73 %) followed by proposed method 2 on ANN (2.5 %) which are close and both very good.

Databases	Details	Training		Testing	
		Face	Voice	Face	Voice
<b>ORL without external effects</b> + <b>Voice</b>	<b>Samples</b>	280x1600	280x1600	120x1600	120x1600
	<b>Fused</b>	280x3200		120x3200	
	<b>Authorized</b>	40 subjects / 7 samples each		40 subjects / 3 samples each	
	<b>Unauthorized</b>	/		160 subjects / 10 samples each	

Table 2: Description of Experiment I databases.

Features		Classifier	RR (%)	EER (%)	Th (%)	AUC
<b>Raw Faces &amp; MFCC + VQ</b>	Proposed Method 1: Concatenated <sup>(pn)</sup>	ANN	95.83	7.5	51	0.9515
		K-NN	89.16	5.24	60	0.719
	Proposed Method 2: Merged <sup>(n)</sup>	ANN	99.1667	<b>2.5</b>	34.4	0.9947
		K-NN	96.66	3.706	60	0.8505
	Proposed Method 3: Multiplied <sup>(n)</sup>	ANN	95.83	14.1667	32.2	0.9137
		K-NN	90	9.237	60	0.7374
<b>PCA for Faces &amp; MFCC+VQ</b>	Concatenated <sup>(n)</sup>	ANN	94.1667	13.32	27.3	0.9351
		K-NN	90.83	3.7125	33.4	0.8905
<b>DCT for Faces &amp; MFCC+VQ</b>	Concatenated <sup>(n)</sup>	ANN	96.667	8.35	33.9	0.9714
		K-NN	96.66	<b>1.73</b>	40	0.923

<sup>(pn)</sup>Pre-normalized features. <sup>(n)</sup>Normalized features.

Table 3: Results with different schemes of fusion and classification.

### 5.2 Experiment II: (ORL with external effects + voice)

We had to introduce some effects in order to enrich the data, because it is a necessity for the neural network to have different and versatile features to enhance the way it learns the variety of appearances and details. Each image had undergone 5 effects thus the 35 samples per subject. As for voice, we took 35 samples of speech for each subject and assigned them to the faces of the corresponding person. A description is in Table 4. The results of this experiment are shown in Table 5 describing the recognitions rates and equal error rates.

In terms of recognition rate, when trained and tested with external effects, ANN dominated K-NN in every fusion scenario, with an average of (97.66 vs 90.99%) with a significant difference ( $p=0.027<0.05$ ). A total test recognition was reached by proposed method 2 (Raw Face & MFCC + VQ merged and normalized) on ANN (100%). The next competing system is the classical method 5 (DCT of Faces & MFCC + VQ) and was on ANN as well (99.37vs 96.26% for K-NN). As for EER, proposed method 2 has attained the lowest error on ANN

(1.67%) against K-NN (8.1 %) which is a very good result, mainly attained by enriching the system by more samples with external effects.

In methods 3, 4 and 5, K-NN performed better than ANN, with an EER of  $13.11 \pm 1.9$  % in average compared to ANN with EER of  $26.91 \pm 7.6$  % with a significant difference ( $p=0.03<0.05$ ). These methods are either highly sensitive to noise where features could be altered significantly, or the neural network configuration was not suitable for this kind of data after effects were involved stating the change of illumination by Gaussian noise as well as eyes cover which can prevent the system from recognizing one’s identity if it depended on those features.

Even though the eigenvectors on Method 4 were sorted in a descending order with respect to their corresponding eigenvalues, this method gave the worst EER on ANN (35.67%), the same thing with Method 5 (23.14%), this is basically related to the unbalance of the system where face features dimensionality was much less than voice features dimensionality (100 vs 1600) and (144 vs 1600) respectively.

Databases	Details	Training		Testing	
		Face	Voice	Face	Voice
<b>ORL with external effects</b> + Voice	<b>Samples</b>	1400x1600	1400x1600	480x1600	480x1600
	<b>Fused</b>	1400x3200		480x3200	
	<b>Authorized</b>	40 subjects / 35 samples each		40 subjects / 12 samples each	
	<b>Unauthorized</b>	/		160 subjects / 45 samples each	

Table 4: Description of Experiment II databases.

	Features	Classifier	RR (%)	EER (%)	Th	AUC
<b>Raw Faces &amp; MFCC + VQ</b>	Proposed Method 1: Concatenated <sup>(pn)</sup>	ANN	94.37	<b>9.02</b>	62.4	0.9989
		K-NN	85.41	10.84	20	0.8570
	Proposed Method 2: Merged <sup>(n)</sup>	ANN	100	<b>1.67</b>	54.1	0.9960
		K-NN	95.62	8.1	20	0.9213
	Proposed Method 3: Multiplied <sup>(n)</sup>	ANN	96.45	21.94	48.7	0.8597
		K-NN	86.45	<b>14.83</b>	40	0.8408
<b>PCA for Faces &amp; MFCC+VQ</b>	Concatenated <sup>(n)</sup>	ANN	98.12	35.67	40.2	0.7056
		K-NN	91.25	<b>13.44</b>	54.6	0.8010
<b>DCT for Faces &amp; MFCC+VQ</b>	Concatenated <sup>(n)</sup>	ANN	99.37	23.14	46.3	0.8480
		K-NN	96.26	<b>11.07</b>	63.7	0.8700

<sup>(pn)</sup>Pre-normalized features. <sup>(n)</sup> Normalized features.

Table 5: Results with different schemes of fusion and classification.

### 5.3 Experiment III: (FEI + voice)

We have downsized the FEI images to 40x40 gray pixels, in order to minimize the amount of calculations as compared to the original colored 640x480x3. The number of subjects is 100. Since the FEI images are varying by degrees from left to right, we decided to take random dispositions for each subject. 10 random positions were taken for each person. As for voice, we took 10 samples of speech for each subject and assigned them to the faces of the corresponding person. Totally, the database contains 1000 samples for training. We took

the remaining 4 images for testing and assigned 4 voice samples to them, this makes 400 samples for testing with authorized subjects (Table 6). From the obtained results in this experiment as shown in (Table 7), all fusion methods were better in recognition on ANN than K-NN in average (94.05 vs 79.65%) with a high significance of ( $p = 0.007 < 0.01$ ), because the network system has fit well the data and generalized to the testing images despite the changes in degrees of rotation from left to right. In terms of EER, we ignore *method 1* from discussion because it reports high errors, proposed *method 3* gives same errors on both classifiers, same as *method 5*. Proposed *method 2* on ANN gave the lowest

Databases	Details	Training		Testing	
		Face	Voice	Face	Voice
<b>FEI + Voice</b>	<b>Samples</b>	1000x1600	1000x1600	400x1600	400x1600
	<b>Fused</b>	1000x3200		400x3200	
	<b>Authorized</b>	100 subjects / 10samples each		100 subjects / 4 samples each	
	<b>Unauthorized</b>	/		100 subjects / 10 samples each	

Table 6: Description of Experiment III databases.

	Features	Classifier	RR (%)	EER (%)	Th (%)	AUC
<b>Raw Faces &amp; MFCC + VQ</b>	Proposed Method 1: Concatenated <sup>(pn)</sup>	ANN	86.5	23.15	3.9	0.8489
		K-NN	74.5	15.75	33.4	0.7060
	Proposed Method 2: Merged <sup>(n)</sup>	ANN	97	<b>9.25</b>	53.4	0.9435
		K-NN	81.25	13	33.4	0.7771
	Proposed Method 3: Multiplied <sup>(n)</sup>	ANN	90.5	19.45	33.1	0.8620
		K-NN	72.25	19.55	20	0.7130
<b>PCA for Faces &amp; MFCC+VQ</b>	Concatenated <sup>(n)</sup>	ANN	97.75	15	24.2	0.9326
		K-NN	79	<b>11.25</b>	14.3	0.7991
<b>DCT for Faces &amp; MFCC+VQ</b>	Concatenated <sup>(n)</sup>	ANN	98.5	13.75	39.3	0.9395
		K-NN	91.25	13.2	42.9	0.8173

<sup>(pn)</sup>Pre-normalized features. <sup>(n)</sup> Normalized features.

Table 7: Results with different schemes of fusion and classification.

	Method 1	Method 2	Method 3	Method 4	Method 5
<b>ORL&amp; Voice</b>	0.2325	0.1442	0.1763	0.044	0.0484
<b>ORL with effects &amp;Voice</b>	0.1419	0.0747	0.0189	-0.095	-0.022
<b>FEI&amp; Voice</b>	0.1429	0.1725	0.149	0.1335	0.1222

Table 8: AUC differences for Experiments I, II, III.

EER (9.25%) against the best EER of K-NN on *method 4* (11.25%), the difference is insignificant however ANN was better.

This may be related to the way ANN learns from features containing rotation contrary to K-NN which is a bare distance computation within a predefined radius. The area under the ROC Curve is a good measure of the system performance, basically, the greater is the area the greater is the ratio TPR/FPR meaning a capability to get more correct classification for less incorrect ones (1 is the maximum value). The Table 8 shows the differences between AUCs of ANN and KNN (subtracting AUC of KNN from the AUC of ANN) for each fusion method. The results have been obtained from the three experiments for the three virtual databases. It is noticeable that most differences are positive. If the used virtual bases are considered separately, ANN is better than KNN in at least 2 out of 5 results. Considering an analysis based on each fusion method, ANN gave better results than KNN in at least 2 out of 3 results. Finally, taking all methods and bases into account, ANN outperformed KNN (13 out of 15). These observations lead to the deduction that ANN has an undeniable (outstanding) potential to perform better than KNN for all experimented fusion methods.

### 5.4 Experiment IV: (ORL + Voice) on PCA

In this Experiment, the idea is to train the database without external effects and test it with effects. In order

to avoid the curse of dimensionality and have some flexibility in the training, as well as avoiding the system unbalance found in Experiment II for Method 4 and 5, we used PCA for the whole database Raw Faces & Voice with features normalized and merged because it was found to be the best system in the previous Experiments I & II& III. The major aim of this experiment is to evaluate the response to noise and external effects. The results of this experiment are tabulated in Table 9. A comparison of the recognition rates and EERs with and without effects between ANN and K-NN is tabulated in Table 10. In an intra-classifiers comparison of recognition rates, it is remarkable that external effects and noise have affected ANN with a high significance (a drop of 10%), but still behaved better than K-NN (a drop of 20%).

In inter-classifiers comparison, ANN outperformed K-NN with and without effects significantly as well. As for EERs, the error rates have increased significantly in both classifiers when noise was involved, (2.5 vs 20.48% ANN and 3.68 vs 14.3% K-NN). Even though there is an insignificant difference in averages between ANN and K-NN (20.48 vs 14.3 %), K-NN still reached a low error rate of (10.12%) while ANN kept a high EER (19.37%). For neural networks, this is an under fitting problem where the network is highly biased and generalizes too much to the point of reaching a high uncertainty whether to accept authentic subjects or reject imposters. This problem can be approached by tuning the network with other parameters as will follow in the next section proceeding.

	RR(%)		EER (%)		Eigenvectors
	No effects	Effects	No Effects	Effects	
ANN	99.16	87.9	2.5	<b>22.7</b>	80 eig
K-NN	95	76.45	3.01	<b>20.69</b>	
ANN	99.12	89.58	2.5	<b>19.37</b>	200 eig
K-NN	95.38	74.58	4.36	<b>12.09</b>	
ANN	99.16	89.97	2.5	<b>19.37</b>	280 eig
K-NN	96.66	76.04	3.67	<b>10.12</b>	

Table 9: Comparison between ANN and K-NN tested with and without effects.

		No effects	Effects	Significance
RR (%)	ANN	<b>99.14</b>	<b>89.15</b>	$p = 9.52 \cdot 10^{-5} < 0.001$
	K-NN	95.68	75.69	$p = 1.23 \cdot 10^{-5} < 0.001$
	Significance	$p = 0.002 < 0.01$	$p = 9.37 \cdot 10^{-5} < 0.001$	/
EER (%)	ANN	<b>2.5</b>	20.48	$p = 8.5 \cdot 10^{-5} < 0.001$
	K-NN	3.68	<b>14.3</b>	$p = 0.03 < 0.05$
	Significance	$p = 0.03 < 0.05$	$p = 0.14 > 0.05$ (NS)	/

Table 10: Comparison of average RR% intra and inter classifiers with and without effects.

Neural Networks	RR (%)		EER (%)		$\lambda$
	No effects	Effects	No effects	Effects	
ANN1	99.16	<b>91.25</b>	2.5	18.12	1
ANN2	99.16	<b>93.75</b>	2.5	14.79	0.1
ANN3	99.16	<b>94.58</b>	1.66	11.25	0.01
ANN4	99.16	<b>95.41</b>	1.66	<b>9.1</b>	0.001
ANN5	99.16	<b>95.62</b>	1.66	<b>7.7</b>	0.0001
ANN6	99.16	<b>96.45</b>	1.66	<b>8.95</b>	0.00001
ANN7	99.16	<b>96.87</b>	1.66	<b>5.83</b>	<b>0.000001</b>

Table 11: Results of tuning the neural network when tested with and without effects.

	Before Tuning (with effects)	After Tuning (with effects)	Significance
RR (%)	89.15	<b>94.84</b>	$p = 4.22 \cdot 10^{-6} < 0.001$
EER (%)	20.48	<b>10.82</b>	$p = 6.61 \cdot 10^{-6} < 0.001$

Table 12: Comparison of recognition rates and EERs pre and post tuning when testing with effects in average.

### 5.5 Experiment V: dependency of the neural network

In order to assess the dependency of the system either on face or voice or both of them, and to avoid the problem of over fitting as well as under fitting, we designed some more complex systems containing from 1 to 4 hidden layers and tested them with black faces (Faces features= 0), white faces (Faces features = 1) and without voice (Voice features = 0). A description of the configurations is in test 1 (Tables 13) and test2 (Table 14). Since recognition rates were low in test 1, we tried to change the configurations in order to confirm the results. In the

both tests, when faces were made black, recognition rates have dropped to averages of  $4.69 \pm 2.55\%$  and  $4.69 \pm 2\%$ . This is because a great number of zeros in the test features will zero so many connection weights in the prediction model by multiplication which affects significantly the recognition.

For white faces, rates were much better than with black faces  $8.35 \pm 3.66\%$  ( $p < 0.001$ ) and  $8.54 \pm 4.35\%$  ( $p < 0.001$ ) for test 1 and 2 respectively, this confirms the first hypothesis. However without voice, accuracies were high in layer 1 (test1 gave  $55.5 \pm 4.5\%$ , test2 gave  $55.5 \pm 4.05\%$ ). We understood that neural networks were relying on face features more than voice features. This can be related to the difference of ranges and variances

#### Test 1

1 Layer	Input	Hidden Layers				Output
	280	500				40
2 Layers	Input	Layer 1	Layer 2		Output	
	280	250	250		40	
3 Layers	Input	Layer 1	Layer 2	Layer 3	Output	
	280	200	150	150	40	
4 Layers	Input	Layer 1	Layer 2	Layer 3	Layer 4	Output
	280	200	100	100	100	40

Table 13: Characteristics of 4 complex configurations of neural networks in terms units.

#### Test 2

Since recognition rates were low in test 1, we tried to change the configurations in order to confirm the results.

1 Layer	Input	Hidden Layers				Output
	280	500				40
2 Layers	Input	Layer 1	Layer 2		Output	
	280	500	300		40	
3 Layers	Input	Layer 1	Layer 2	Layer 3	Output	
	280	500	300	100	40	
4 Layers	Input	Layer 1	Layer 2	Layer 3	Layer 4	Output
	280	500	400	300	200	40

Table 14: Characteristics of 4 complex configurations of neural networks in terms of units.

between faces and voices in our database taking into consideration that our normalization rule was not linear. Unfortunately, a homogeneity test was not performed to assess our databases. In the other hand, as the system started containing more hidden layers, the accuracies dropped to the level of faces, this means that the system started leaning from voices same as faces approximately, however the recognition was still bad which is not a good point.

## 5.6 Discussions

- In Experiment I, we used ORL & Voice fused using five different schemes, we trained and tested without external effects. Proposed *method 2* behaved very well on ANN and performed better than others. Proposed *method 3* has given a good recognition rate but is was the faultiest method.
- In Experiment II, we repeated Experiment I introducing external effects in training and testing databases. *Proposed method 2* implemented on ANN gave again the best RR and EER. *Methods (3,4,5)* were unacceptable with ANN contrary to their performance on K-NN.
- In Experiment III, we tested the capability of neural networks to generalize to unseen modals containing degrees of rotations for faces fused with voice. *Proposed method 2* reached the best results in terms of recognition rates and equal error rates. In contrast, *proposed method 3* was totally unacceptable. The AUC analysis was run on both classifiers performing on five methods of the study, with all the previous experiments, and has shown that from this criterion point of view, neural networks were much better than K-NN.
- In Experiment IV, we got back to ORL & voice and applied PCA on the whole database with normalized and merged features. We trained without external effects and tested with noise and effects. This has been done to assess the response to noise when not trained with. In terms of recognition rate, ANN performed well, in contrast with EER where it failed to give a low error. A tuning protocol was set up and applied in order to adapt the system to the type of data and solve the problem encountered consisting of under fitting. This was done mainly by varying the regularization parameters of the networks. The procedure of tuning gave good and promising results and confirmed the flexibility of neural networks.
- In Experiment V, we have done a dependency test on different configurations with a variety of regularization parameters, we found the system to be depending on face features over voice features. We could lower this dependency by designing more complex configurations, however, the recognition rates kept very bad telling that the system could not perform well in absence of one of the modalities.

## 6 Conclusion

In this work, we have introduced the concept of data fusion and explained why Multibiometric systems

perform better than Unimodal systems. Our experimental part contained four experiments mainly done on two virtual databases, ORL & Voice, and FEI & Voice. Throughout Experiments I and II, *proposed method 2* gave the best recognition rates (99.16 and 100 %) and realized the least faulty systems (2.5 and 1.67%). We understand ultimately that ANN trained with merged and normalized data features from different modalities can be very effective. In experiment III where the database was much larger than the first and second trial, recognition rates diminished slightly and the equal error rate has increased significantly (9.25 %) but it maintained its position yielding the best performance since all other schemes have deteriorated as well. Although the *proposed method 1* was relatively good in experiments I and II, it was remarkably defective on the FEI database with Voices (23.11%), we concluded that normalizing features could have a powerful impact on the behavior of the neural network especially when the feature ranges are not approximate. *Proposed method 3* led to the conclusion that multiplying non-homogenous features as face and voice could alter unexpectedly the distinctive characteristics of different classes thus result in a completely unreliable system in comparison to the *proposed method 2*.

It is to mention that the classical methods 4 and 5 involving PCA and DCT for faces and MFCC & VQ for voices were much more effective in K-NN than ANN, this says basically that when features fed to a neural network are dimensionally unbalanced, the performance of the system could drop badly. In contrast with K-NN which is a simple distance measure that would not be affected by this problem. In experiment IV, we showed how neural networks could be influenced by noise and external effects simulating real-life scenarios. This has been done by training without effects and testing with them. Even though the results between K-NN performing better than ANN against noise were insignificant, we decided to set up a diagnosis protocol aiming to approach this problem. This has been done by discovering whether the modal of the neural network was under fitting the data, just well-fitting the data or over fitting it. The problem in hand was under fitting, it was resolved by changing the configurations in a convenient manner (Tuning the network) citing the layers and the regularization parameters. Using this perspective could lead to very promising and adaptive performances.

Finally, it is to be emphasized that we were able to achieve two major purposes of this study, first, was validating an effective data fusion method at feature level (*proposed method 2* merging and normalizing features with equal dimensions), and second, consists of taking a good grasp of the concept of neural networks to the point of controlling its behavior as wanted to achieve good and better results.

As for further works, we hope applying this study on a better database where voices are recorded in an anechoic chamber. Also to apply a homogeneity test on this database in order to have a good statistical understanding of the features being fed to the recognition systems in hand.

## References

- [1] Faundez-Zanuy M. *Data fusion in biometrics*. IEEE A&E Systems Magazine, 20(1):34-38. January, 2005.  
<https://doi.org/10.1109%2Fmaes.2005.1396793>
- [2] Almahafzah H., Imran M., and Sheshadri H.S. *Multi-algorithm Feature Level Fusion Using Finger Knuckle Print Biometric*. In *FGIT-FGCN/DCA*, pp. 302-311, 2012.  
[https://doi.org/10.1007%2F978-3-642-35594-3\\_42](https://doi.org/10.1007%2F978-3-642-35594-3_42)
- [3] Ross A., Jain A.K. *Multimodal Biometrics: an Overview*. Proceeding of 12th IEEE European Signal Processing Conference, pp.1221-1224, Austria, September 2004. ISBN: 978-320-0001-65-7.
- [4] Nada Alay and Heyam H. Al-Baity. *A multimodal biometric system for personal verification based on different level fusion of iris and face traits*. Bioscience Biotechnology Research Communications. publisher by Society for Science and Nature, 12(3):565-576. September 2019.  
<https://doi.org/10.21786%2Fbbrc%2F12.3%2F3>
- [5] Oloyede, M. and Hancke, G. *Unimodal and Multimodal Biometric Sensing Systems: A Review*. IEEE Access; 4:7532--7555, 2016.  
<https://doi.org/10.1109%2Faccess.2016.2614720>
- [6] Fernandes, S.L. & Josemin Bala, G. Analyzing State-of-the-Art Techniques for Fusion of Multimodal Biometrics. Proceedings of the Second International Conference on Computer and Communication Technologies, pp.473–478, 2015.  
[http://dx.doi.org/10.1007/978-81-322-2526-3\\_49](http://dx.doi.org/10.1007/978-81-322-2526-3_49).
- [7] Chetty G. and Wagner M. *Robust face-voice based speaker identity verification using multilevel fusion*. Image and Vision Computing. 26(9): 1249–1260, 2008.  
<https://doi.org/10.1016%2Fj.imavis.2008.02.009>
- [8] Palanivel, S. & Yegnanarayana, B. *Multimodal person authentication using speech, face and visual speech*. Computer Vision and Image Understanding, 109 (1):44–55, 2008.  
<http://dx.doi.org/10.1016/j.cviu.2006.11.013>.
- [9] Raghavendra, R., Ashok Rao, and G. Hemantha Kumar. *Multimodal Person Verification System Using Face and Speech*. Procedia Computer Science 2: 181–187, 2010.  
<https://doi.org/10.1016%2Fj.procs.2010.11.023>
- [10] Elmir Y, Elberrichi Z., Adjoudj R. *Multimodal biometric using a hierarchical fusion of a person 's face, voice, and online signature*. Journal of Information Processing Systems, 10(4): 555-567; 2014.  
<https://doi.org/10.3745/jips.02.0007>
- [11] Soltane M. Figueiredo-Jain (FJ) Tune Algorithm for Gaussian Mixture Modal (GMM) Based Face and Signature Multi-Modal Biometric Verification Fusion Systems. Journal of Computational Intelligence and Electronic Systems. 4(1): 27-36, 2015.  
<https://doi.org/10.1166%2Fjcieis.2015.1110>
- [12] Kasban H. *A robust multimodal biometric authentication scheme with voice and face recognition*. Arab Journal of Nuclear Sciences and Applications 50(3):120–130, 2017. ISSN 1110-0451.
- [13] Gad R, El-Fishawy N, El-Sayed A, Zorkany M. *Multi-biometric systems: a state of the art survey and research directions*. International Journal of Advanced Computer Science and Applications, 6(6):128-138, 2015.  
<https://doi.org/10.14569%2Fijacsa.2015.060618>
- [14] Balaka Ramesh Naidu, P.V.G.D Prasad Reddy. *Fusion of Face and Voice for a Multimodal Biometric Recognition System*. International Journal of Engineering and Advanced Technology (IJEAT), 8(3), February 2019. ISSN: 2249–8958.
- [15] Shang-Hung Lin. "An introduction to face recognition technology". Informing Science: The International Journal of an Emerging Transdiscipline, (3):001-007, 2000.  
<https://doi.org/10.28945%2F569>
- [16] Cherifi D., Radji N., Nait Ali A., "Effect of Noise, Blur And Motion On Global Appearance Face Recognition Based Methods Performance", International Journal of Computer and Applications, 16(6):0975–8887, February 2011.  
<https://doi.org/10.5120/2019-2723>
- [17] Draper B.A., Baek K., Bartlett M.S., and Beveridge J.R. "Recognizing Faces with PCA and ICA". Computer Vision and Image Understanding, 91(1-2):115-137, 2003.  
[https://doi.org/10.1016/s1077-3142\(03\)00077-8](https://doi.org/10.1016/s1077-3142(03)00077-8)
- [18] Turk M. and Pentland A., "Face recognition using eigenfaces". Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 586-591, Maui, HI, USA, June, 1991.  
<https://doi.org/10.1109/cvpr.1991.139758>
- [19] Turk M. and Pentland A. *Eigenfaces for recognition*. Journal of Cognitive Neuroscience, 3(1): 71-86, 1991.  
<https://doi.org/10.1162/jocn.1991.3.1.71>
- [20] Chen Y. and Zhao Y. *Face Recognition Using DCT and Hierarchical RBF Model*. Intelligent Data Engineering and Automated Learning (IDEAL), pp. 355–362, 2006.  
[https://doi.org/10.1007%2F11875581\\_43](https://doi.org/10.1007%2F11875581_43)
- [21] Nagil J., Khaleel Ahmed S. and Nagi F. *Pose Invariant Face Recognition using Hybrid DWT-DCT Frequency Features with Support Vector Machines*. In Proceedings of the 4th International Conference on Information Technology and Multimedia at UNITEN (ICIMU), Malaysia, pp. 99-104, November 2008.  
<https://www.researchgate.net/publication/228699251>
- [22] Hajiababi M., Askari J., Sadri S., and Saraee M. *Face Recognition using Discrete Cosine Transform plus Linear Discriminant analysis*. Proceedings of the World Congress on Engineering, I:652-655, London, U.K, July 2007.

- [http://www.iaeng.org/publication/WCE2007/WCE2007\\_pp652-655.pdf](http://www.iaeng.org/publication/WCE2007/WCE2007_pp652-655.pdf)
- [23] Reynolds A.D. *An Overview of automatic speaker recognition technology*. Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 4:4072–4075, Orlando, FL, USA, 2002. <https://doi.org/10.1109%2Ficassp.2002.5745552>
- [24] Majeed, S. A., Husain, H., Samad, S. A., and Idbeaa, T. F. Mel Frequency Cepstral Coefficients (MFCC) Feature Extraction Enhancement in the Application of Speech Recognition: a Comparison Study. *Journal of Theoretical and Applied Information Technology* 79(1):38-56, September 2015. <http://www.jatit.org/volumes/seventynine1.php>
- [25] Kaur M., Girdhar A., Kaur M. *Multimodal biometric system using speech and signature modalities*. *International Journal of Computer Applications*. 5(12):13-16, August 2010. <https://doi.org/10.5120%2F962-1339>
- [26] Elmir Y., Elberrichi Z., Adjoudj R. *A hierarchical fusion strategy based multimodal biometric system*. The International Arab Conference on Information Technology (ACIT'2013). Khartoum, December 17-19, 2013. <https://doi.org/10.13140/RG.2.1.4675.1842>
- [27] Illinois Image Formation and Processing (IIFP). *DSP mini-project: An automatic Speaker Recognition System*. [http://minhdo.ece.illinois.edu/teaching/speaker\\_recognition](http://minhdo.ece.illinois.edu/teaching/speaker_recognition)
- [28] Cherifi D., Hafnaoui I., Nait Ali A., "Multimodal Score-Level Fusion Using Hybrid GA-PSO for Multibiometric System", *International journal of computing and informatics*, 39(1): 209–216, 2015. <http://www.informatica.si/index.php/informatica/article/download/837/622>
- [29] Liestol K., Anderson PK., Anderson U., *Survival analysis and neural nets*, *Statistics in Medicine*, 13(12):1189-1200, June 1994. <https://doi.org/10.1002%2Fsim.4780131202>
- [30] Andrew Ng, "Machine Learning Online Course". University of Stanford, 2011. <https://freevidelectures.com/course/2257/machine-learning>.
- [31] Williams D. and Hinton G. *Learning representations by back-propagating errors*. *Nature*, 323(6088): 533-538, October 1986. <https://doi.org/10.1038%2F323533a0>
- [32] Hinton GE. *How neural networks learn from experience*. *Scientific American* 267(3):145–151, September 1992. <https://doi.org/10.1038%2Fscientificamerican0992-144>
- [33] Ba Lathika, D Devaraj. *Artificial Neural Network Based Multimodal Biometrics Recognition System*. International Conference On Control, Instrumentation, Communication and Computational Technologies (ICCCCT). Kanyakumari, India 10-11 July 2014. <https://ieeexplore.ieee.org/document/6993100>
- [34] Soleymani s. , Dabouei A., Kazemi H., Dawson J. and Nasrabadi N. M., *Multi-Level Feature Abstraction from Convolutional Neural Networks for Multimodal Biometric Identification*, 24th International Conference on Pattern Recognition (ICPR), pp. 3469-3476, Beijing, 2018. <https://doi.org/10.1109/icpr.2018.8545061>

# Comparison of the Community Structure Partition Optimization of Complex Networks with Different Community Discovery Algorithms

Renjie Peng and Yunxia Yao

Information Engineering Department, Longdong University, Qingyang, Gansu 745000, China

E-mail: yunxyao@126.com

## Student paper

**Keywords:** community structure partition, complex network, density peak clustering, common neighbor node

**Received:** December 23, 2019

*Complex problems can be transformed into complex networks. Through the community partition of complex networks, the relationship between nodes can be found more clearly. This paper briefly introduces three algorithms for community structure partition of complex networks, which were based on the similarity of common neighbor nodes, ant colony algorithm and density peak clustering, and compared the performance of the three algorithms by using six artificial networks whose chaotic factors gradually increased as well as two real networks in MATLAB software. The results suggested that the increase of chaotic factors in the artificial network reduced the normalized mutual information (NMI) of the partition results calculated by the three algorithms. However, the NMI of the algorithm based on density peak clustering in the same artificial network was the highest, the algorithm based on ant colony algorithm followed, and the algorithm based on the similarity of common neighbor nodes performed the worst. For a real example network, the modularity of the algorithm based on density peak clustering was the highest, the algorithm based on ant colony algorithm was the second, and the algorithm based on the similarity of common neighbor nodes was last. In conclusion, the fuzzier the community structure is in the complex network, the lower the performance of the partition algorithm is, and the algorithm based on density peak clustering has the best performance.*

*Povzetek: Podana je primerjava razčlenitve strukture kompleksnih omrežij s pomočjo algoritmov za odkrivanje.*

## 1 Introduction

Real life is composed of many complex problems. Studying the laws that govern them can help solve them, which can in turn promote social development [1]. However, the complexity of real-life problems makes the surface laws that can be found intuitively have little value, and those surface laws have no fundamental impact on the solution of problems. In order to mine the hidden information, the complex problem is transformed into a complex network. Nodes in a complex network represent the individuals participating in it, and line segments between nodes represent the relationships between them [2]. For example, the rapid development of the Internet can be seen as a complex network. The participating users using the Internet have different identities. They will search the information on Internet according to their own interests and hobbies. Users with different identities will gradually focus on the same or similar interests, thus forming a community structure [3]. Similarly, the structure of a protein can also be seen as a complex network, where genes are nodes. Through community division, genes with similar functions can be summarized, so as to mine the effective information of genes. Through the division of community structure in the complex network, the information contained in the complex network can be more clearly understood, so as to analyze

the topological properties and organizational structure of the complex system. Liu et al. [4] put forward a weighted maximum fitness algorithm which optimized the initial node according to the potential energy idea, simplified the node fitness function, and expanded the community according to the potential energy queue. The simulation results showed that the algorithm had higher accuracy and shorter calculation time than the maximum fitness algorithm. Zuo et al. [5] detected the similar energy behavior nodes in the sensor node network using complex network community division algorithm and selected the cluster center and hop nodes using the immune response principle, so as to realize the energy-saving topology of the sensor network. The experimental results showed that the method could reduce the energy consumption. Huang et al. [6] put forward a software network optimal partition method based on the dependency between software functions and verified through experiments that the method could effectively detect the optimal communities in various software. This paper briefly introduces three algorithms for the community structure division of complex networks, which were based on the similarity of common neighbor nodes, ant colony algorithm and density peak clustering algorithm, and compared the performance of the three algorithms by using six artificial

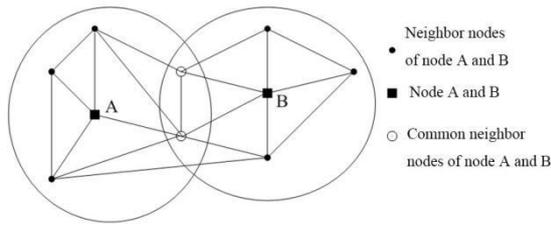


Figure 1: Similarity of common neighbor nodes.

networks whose chaotic factors gradually increased and two real networks in MATLAB software.

## 2 Multiple community partition algorithms

### 2.1 Community partition based on the similarity of common neighbor nodes

There are many community partition algorithms that can be used in complex networks. Firstly, the community partition algorithm based on the similarity of common neighbor nodes is introduced. The schematic diagram is shown in Figure 1. Black square point A is any point in community A, black square point B is any point in community B, black circle points are the adjacent node of node A and node B, and hollow circle points are the common adjacent node of node A and node B, then the relationship of nodes in Figure 1 [7] can be expressed as:

$$\begin{cases} H(A) = I(A, B) + H(A|B) \\ H(B) = I(A, B) + H(B|A) \\ H(A, B) = H(A) + H(B|A) \\ H(A, B) = H(B) + H(A|B) \end{cases}, (1)$$

where  $H(A)$  and  $H(B)$  represent the node sets belonging to community A and B,  $I(A, B)$  represents the common node set of community A and B,  $H(A|B)$  represents a set of nodes belonging to community A but not belonging to community B,  $H(B|A)$  represents a set of nodes belonging to community B but not belonging to community A. According to the above relationship model, the similarity of two nodes can be calculated:

$$S_{a,b} = \frac{I(A, B)}{\max(H(A, H(B)))}, (2)$$

where  $S_{a,b}$  stands for the similarity of node a and b.

The flow of community partition algorithm based on the similarity of common neighbor nodes is shown in Figure 2.

- ① First, the complex network  $G = (V, E)$  to be divided is input, where  $V$  is a set of nodes in the complex network and  $E$  is a set of segments between nodes.
- ② According to equation (2), the similarity matrix of the common neighbor nodes is calculated. In the initial network, each node is regarded as a community, which

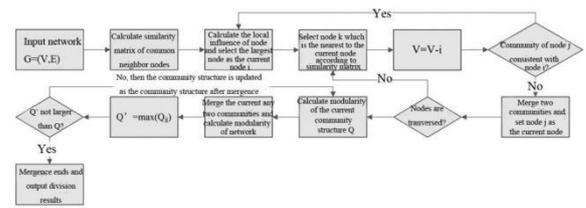


Figure 2: The process of community partition based on the similarity of common neighbor nodes.

gradually aggregates into different communities in the following iteration.

- ③ The local influence of each node in the network is calculated [8]:

$$L_i = \sum_{j \in \Gamma_i} \sum_{v \in \Pi_j} N_v, (3)$$

where  $L_i$  stands for the local influence of node  $i$  in the network,  $\Gamma_i$  is the set of neighbor nodes of node  $i$ ,  $\Pi_j$  is the secondary neighbor node set of node  $i$ , and  $N_v$  is the degree of the secondary neighbor node of node  $i$ . The node with the largest local influence is selected from the network and set as current node  $i$ .

- ④ Node  $j$  with the highest similarity with current node  $i$  is selected from the similarity matrix of the common neighbor nodes, and then node  $i$  is removed from node set  $V$  used for calculation to represent that the node has been checked.

- ⑤ The communities to which current node  $i$  and highest similarity node  $j$  belong to are determined. If they are the same, step ③ is repeated to select the new current node from the remaining undetected nodes. If they are not the same, the communities of the two nodes are merged, and node  $j$  is set as current node  $i$ .

- ⑥ Whether the nodes in the network are traversed is determined, and step ③ ~ ⑤ are repeated if they are not traversed; after traversing, modularity  $Q$  of the current network community structure is calculated.

- ⑦ After merging any two communities in the current network, the modular degree of the community structure is calculated after merging, and then the merging scheme with the largest modular degree is selected. Modular degree  $Q'$  under the selected scheme is compared with modular degree  $Q$  before merging. If  $Q'$  is greater than  $Q$ , the network community structure is updated according to the merging scheme, and it is repeated until  $Q'$  is not greater than  $Q$ . Finally the result is output.

### 2.2 Community partition based on the ant colony algorithm

In addition to the common neighbor similarity method described above, the ant colony algorithm can also be used to divide communities in complex networks. In the community partition of complex network, the ant colony algorithm [9] regards the nodes in the network as ants and then imitates the moving form of ants in foraging to transfer the location of network nodes, so as to realize the node aggregation and community partition.

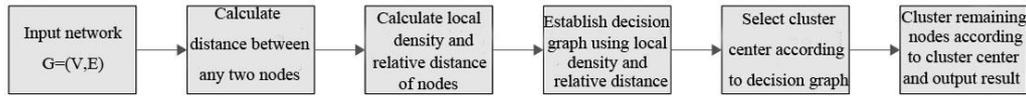


Figure 4: Community partition process based on density peak clustering.

The flow of community division based on the ant colony algorithm is shown in Figure 3.

- ① Complex network  $G = (V, E)$  to be divided is input.
- ② The ant position is initialized: motif  $u_i$  which is composed of core node and neighbor ordinary nodes is labeled as  $NP_i$   $i = (1, 2, 3, \dots, k)$  using string encoding. If  $WCC(u_{i+1}, NP_{i+1}) \geq WCC(u_i, NP_i)$ , the two motifs are merged as one community; otherwise  $NP_{i+1}$  is independent as a community until all the merging is over. Then the nodes in the network are mapped as ants in the ant colony algorithm:

$$WCC(u, NP) = \begin{cases} \frac{t(u, NP)}{t(u, V)} \cdot \frac{vt(u, V)}{|NP \setminus u| + vt(u, V \setminus NP)} & t(u, V) > 0 \\ 0 & otherwise \end{cases}, (4)$$

where  $t(u, NP)$  and  $t(u, V)$  stand for the number of triangular motifs which are composed of motif  $u$  and node set NP and the number of triangular motifs which are composed of motif  $u$  and node set V,  $vt(u, V)$  stands for the number of motif  $u$  and node set V which are needed for constituting at least one triangular motif,  $vt(u, V \setminus NP)$  stands for the number of motif  $u$  and node set V with NP node set removed which are needed for constituting at least one triangular motif, and  $|NP \setminus u|$  stands for the number of NP node set after removing motif  $u$ .

- ③ The ants (nodes) in the initially divided network move to the adjacent motif according to the probability transfer formula, and the probability transfer formula [10] is:

$$P(i, j) = \begin{cases} \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{v_q \in N_i} (\tau_{iq})^\alpha (\eta_{iq})^\beta} & v_j \in N_i \\ 0 & otherwise \end{cases}, (5)$$

where  $v_j$  stands for the neighbor node of node  $v_i$ ,  $v_q$  stands for all the neighbor nodes of node  $v_i$ ,  $\tau_{ij}$  and  $\tau_{iq}$  stand for the heuristic functions between  $v_i$  and  $v_j$  and between  $v_i$  and  $v_q$ , and  $\eta_{ij}$  and  $\eta_{iq}$  stand for the pheromones of the path between  $v_i$  and  $v_j$  and between  $v_i$  and  $v_q$ .

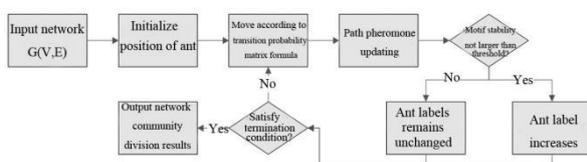


Figure 3: The community partition process based on ant colony algorithm.

- ④ After the ant is moved, the pheromone of the path is updated [11], and the update formula is:

$$\begin{cases} \tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{p=1}^m \Delta\tau_{ij}^p \\ \Delta\tau_{ij}^p = Q \end{cases}, (6)$$

where  $\rho$  is the evaporation rate of pheromone,  $m$  is the number of ants, and  $Q$  is the probability that the  $p$ -th ant chooses the path.

- ⑤ The WCC ratio of ants before and after moving is calculated according to equation (4). If the ratio is not greater than the set threshold, the label of ants will increase (the node will be added to the target module); otherwise, the label of ants will remain unchanged (the node still belongs to the original module).

- ⑥ Whether the algorithm meets the termination condition is determined: after the maximum number of iterations is reached, steps ③ ~ ⑤ are repeated if the termination condition is not satisfied; the partition result is output if the termination condition is satisfied.

### 2.3 Community partition based on density peak clustering

The basic principle of community partition based on density peak clustering is to select cluster center nodes from the complex network and then assign non-cluster center nodes to different cluster center nodes according to the distance between nodes, so as to achieve the effect of community partition. The criteria for judging whether a node is a cluster center are: ① the local density of the node itself is large enough; ② the distance between the node and other nodes with sufficient density is large enough. The formula [12] for calculating the local density and relative distance of nodes in a complex network is as follows:

$$\begin{cases} \rho_i = \sum_j \exp(-\frac{d_{ij}^2}{d_c^2}) \\ \delta_i = \begin{cases} \max_j \{d_{ij}\} & \rho_i = \max\{\rho\} \\ \min_{j:\rho_i < \rho_j} \{d_{ij}\} & otherwise \end{cases} \end{cases}, (7)$$

where  $\rho_i$  is the local density of node  $i$ ,  $\delta_i$  is the relative distance of node  $i$ ,  $d_{ij}$  is the distance between node  $i$  and  $j$ , and  $d_c$  is the cutoff distance, i.e., the search range of node density.

As shown in Figure 4,

- ① complex network  $G = (V, E)$  to be divided is input first.

② the distance between any two nodes in the complex network is calculated. In the complex network, there may not only be one path between any two nodes, so it will affect the calculation of node distance. In this study, the equivalent resistance theory is used to integrate the multi-path between nodes, and the node distance is calculated [13]:

$$d_{ij} = \ln\left(\frac{1}{\sum_{n=1}^m \frac{1}{\exp(P_n) - 1}} + 1\right), \quad (8)$$

where  $P_n$  is the length of the  $n$ -th path between node  $i$  and  $j$  and  $m$  is the number of valid paths between nodes. As mentioned above, the path between nodes in a complex network is mostly more than one. If each path is regarded as a "resistance", the circuit formed by the network will have the problem of interleaving of parallel connection and series connection, which greatly increases the difficulty of calculation. Therefore, based on the small world effect of a complex network, the path with a small amount of information with a distance greater than 2 is eliminated, so as to reduce the amount of calculation and avoid the influence of interleaving of parallel connection and series connection.

③ The local density and relative distance of each node in the network is calculated according to equation (7), and then the decision diagram of node is drawn by taking the local density and relative distance as the horizontal and vertical coordinates. Each node in the decision diagram will present the situation of their own aggregation according to the nearness degree of horizontal and vertical coordinates. Generally speaking, non-cluster center nodes will gather, and the scattered nodes can be regarded as cluster centers.

④ After drawing the decision-making graph of the nodes, the non-clustered and scattered nodes can be identified as cluster centers. Usually, the cluster centers are determined manually, but the efficiency is too low, and the subjectivity is too strong. If the nodes are clustered clearly, the problem is not serious, but if the degree of clustering is low, subjectivity bias is too strong, which is not conducive to community partition. Therefore, the cluster center in the decision graph is solved using Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm in this study. DBSCAN algorithm is a clustering algorithm based on density. Its calculation steps are as follows. Firstly, an unprocessed point is selected from the decision diagram, and other points within the scanning radius range of the point are searched. If the number of other points exceeds the set minimum number, that point and the other points get together to form a cluster, and the point is labeled as "having been processed", and then the other points in the cluster are processed in the same way to expand the cluster until all the points in the cluster are processed. If the number of points does not reach the set minimum number, that point is labeled as a noise point, and the above steps are repeated until all the points are processed. The noise points, i.e., clustering center nodes used for density peak clustering, are output.

⑤ Taking the cluster center as the standard, the rest nodes are assigned to the nearest cluster, and the community partition results are output finally.

### 3 Simulation experiment

#### 3.1 Experimental environment

In this study, the three algorithms were compiled using MATLAB software, and simulation experiments were carried out. The experiments were carried out in the laboratory server with configurations of Windows 7, Core i7 CPU, and 16 GB memory.

#### 3.2 Experimental data

In this study, LFR tool was used to generate artificial network data sets. The relevant parameters are shown in Table 1. Six artificial network data sets were constructed. The total number of nodes ( $N$ ) in each artificial network data set was 2000, the average nodal degree  $k$  was 30, the maximum nodal degree was 60, the nodal degree distribution parameter  $t1$  was 2, the community size distribution parameter  $t2$  was 1, the maximum size of community was 150, and the minimum size of community was 25. The difference between the six artificial network data sets lied in the blend factor  $mu$ , which represents the probability that a node in a community was connected to an external community node, which increases gradually from 0.2 to 0.7.

In order to further test the effectiveness of the three community partition algorithms, in addition to the artificial network data set, the real network data set was also used for simulation experiments. As shown in Table 2, football refers to the American college football network, and email refers to the email communication network of University of Spain. The above two networks are common real network data sets used for detecting the performance of community partition algorithm. Among them, the nodes of the football network represents a team, totally 115 nodes, and the edges represents having

Network number	LF R1	LF R2	LF R3	LF R4	LF R5	LF R6
N	2000					
k	30					
Max k	60					
t1	2					
t2	1					
Max C	150					
Min C	25					
mu	0.2	0.3	0.4	0.5	0.6	0.7

Table 1: Related parameters of artificial network data set.

Real network data set	football	email
Total number of nodes	115	1133
Number of edges	616	5451
Labeled network or not	yes	no

Table 2: Related parameters of real network data sets.

competed with other teams, totally 616 edges. Moreover, the network belongs to the labeled network, each team belongs to a clear union, that is, the community structure is known. The nodes of the email network represents university students in the network, totally 1133 nodes, and the edges represent having communicated, totally 5451 edges. Furthermore, the network belongs to the unlabeled network. Due to privacy protection, the personal information and social contact of users in the network are unknown, that is, the community structure is unknown.

### 3.3 Algorithm performance evaluation index

Normalized mutual information (NMI) [14] can be used to compare the similarity between the actual community structure with the community structure divided by the algorithm, and its calculation formula is:

$$NMI = 1 - \frac{H(X|Y) + H(Y|X)}{2}, \quad (9)$$

where  $X$  represents the actual community structure,  $Y$  represents the community structure of algorithm partition,  $H(X|Y)$  represents the normalized conditional entropy of structure  $X$  under partition structure  $Y$ , and  $H(Y|X)$  represents the normalized conditional entropy of structure  $Y$  under partition structure  $X$ .

An artificial complex network is a set of network data constructed artificially by the LFR tool, so its actual community partition is known. The performance of the algorithm can be measured by NMI, but for the real-life network, the actual community partition is often unknown, such as the email network used in the experiment in this study. Therefore the partition effect of the unlabeled network by the algorithm is evaluated using modularity [15]:

$$Q = \frac{\sum_{ij} (A_{ij} - \frac{k_i k_j}{2m}) \sigma(c_i, c_j)}{2m}, \quad (10)$$

where  $A_{ij}$  is adjacency matrix,  $k_i$  and  $k_j$  are the degrees of nodes  $i$  and  $j$  respectively, and  $m$  is the total number of network edges. The value of  $\sigma(c_i, c_j)$  is 1 when node  $i$

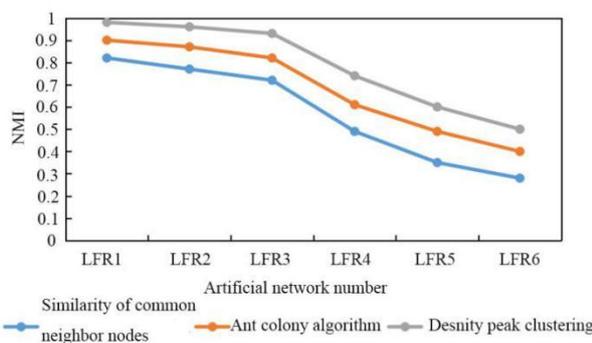


Figure 5: Community partition effects of three algorithms on different artificial networks.

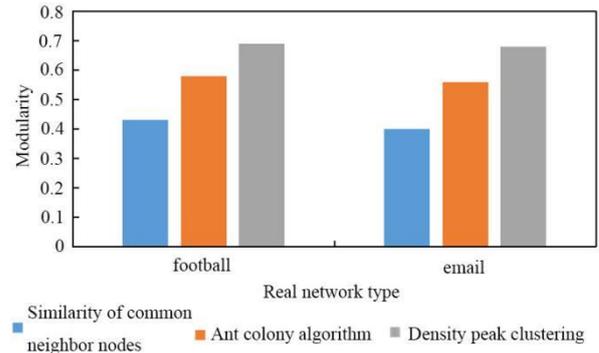


Figure 6: The modularity of the three algorithms for community partition of two real networks.

and  $j$  belong to the same community; otherwise the value is 0.

### 3.4 Experimental results

As shown in Figure 5, the NMI of the community partition algorithm based on the similarity of common neighbor nodes for LFR1~LFR6 artificial network was 0.82, 0.77, 0.72, 0.49, 0.35, and 0.28 respectively; the NMI of the community partition algorithm based on the ant colony algorithm for LFR1~LFR6 artificial network was 0.9, 0.87, 0.82, 0.61, 0.49, and 0.4 respectively; the NMI of the community partition algorithm based on density peak clustering for LFR1~LFR6 was 0.98, 0.96, 0.93, 0.74, 0.6 and 0.5 respectively. In this study, the artificial network LFR1~LFR6 improved the blend factor  $\mu$  gradually, the network community structure became fuzzy gradually, and the NMI of the three algorithms for community structure partition gradually reduced; moreover, in the process of  $\mu$  increase, under the same artificial network, the community partition algorithm based on density peak clustering had the largest NMI, followed by the community partition algorithm based on the ant colony algorithm and the community partition algorithm based on the similarity of common neighbor nodes.

As mentioned above, due to the large number of participating nodes, real networks have complex connections with each other. When complex networks are analyzed using the community partition algorithm, it is to obtain effective information from the unknown complex networks. Therefore, the community structure of real networks is often unknown. NMI is based on the actual structure of the known network, so it is not suitable for the evaluation on the partition effect of real networks by algorithm. In this study, the stability of the community structure partition by the algorithm was measured by modularity, so as to evaluate the partition effect of the algorithm. As shown in Figure 6, for the football network, the community structure modularity of the community partition algorithm based on the similarity of common neighbor nodes was 0.43, that of the community partition algorithm based on the ant colony algorithm was 0.58, and that of the community partition algorithm based on density peak clustering was 0.69. For the email network, the community structure modularity of the community partition algorithm based on the similarity of common

neighbor nodes was 0.4, that of the community partition algorithm based on the ant colony algorithm was 0.56, and that of the community partition algorithm based on density peak clustering was 0.68. The algorithm based on density peak clustering had the highest modularity of community structure, followed by the ant colony algorithm and the algorithm based on the similarity of common neighbor nodes. That is to say, the community structure was the most stable and most similar to the actual structure when the real network was divided by the algorithm based on density peak clustering.

## 4 Conclusion

In this study, three algorithms for the community structure partition of complex networks were introduced, which were based on the similarity of common neighbor nodes, the ant colony algorithm and the density peak clustering algorithm. Then, the performance of the three algorithms was compared by using six artificial networks whose chaotic factors gradually increased and two real networks in MATLAB software. The results are as follows. (1) With the increase of the chaotic factors in the artificial network, the NMI of community partition of the three algorithms gradually decreased, but the NMI of the algorithm based on density peak clustering in the same artificial network was the highest, the algorithm based on the ant colony algorithm was the second highest, and the algorithm based on the similarity of the common neighbor nodes was the lowest. (2) In the real network, the partition performance of the algorithms was evaluated by the modularity, among which the algorithm based on density peak clustering still had the highest modularity, the algorithm based on the ant colony algorithm was the second highest, and the algorithm based on the similarity of common neighbor nodes was the lowest in modularity.

## References

- [1] Qu Y, Guan X, Zheng Q, Liu T, Wang L, Hou Y, Yang Z (2015). Exploring community structure of software Call Graph and its applications in class cohesion measurement. *Journal of Systems & Software*, pp. 193-210. <https://doi.org/10.1016/j.jss.2015.06.015>
- [2] Hong Y, Fu C, Huang QF, Fang ZC, Zeng JH, Han LS (2016). Communities Evolution Analysis Based on Events in Dynamic Complex Network. *IEEE Intl Conf on Dependable, Autonomic & Secure Computing, Intl Conf on Pervasive Intelligence & Computing, Intl Conf on Big Data Intelligence & Computing & Cyber Science & Technology Congress. IEEE*, 97
- [3] Giustolisi O, Ridolfi L (2015). A novel infrastructure modularity index for the segmentation of water distribution networks. *Water Resources Research*, pp. 7648-7661. <https://doi.org/10.1002/2014WR016067>
- [4] Liu SS, Wang H (2015). Community discovery algorithm based on potential energy in complex network. *Control Conference. IEEE*, 7259812
- [5] Chen Z, Jia M, Wang Y, Yan X (2015). A Security Topology Protocol of Wireless Sensor Networks Based on Community Detection and Energy Aware. *2015 IEEE Trustcom/BigDataSE/ISPA. IEEE*, 519
- [6] Huang G, Zhang P, Zhang B, Yin T, Ren J (2016). The optimal community detection of software based on complex networks. *International Journal of Modern Physics C*, pp. 381. <https://doi.org/10.1142/S0129183116500856>
- [7] Yu H, Yao N, Wang T, Li G, Gao Z, Tan G (2016). WDFAD-DBR: Weighting depth and forwarding area division DBR routing protocol for UASNs. *Ad Hoc Networks*, pp. 256-282. <https://doi.org/10.1016/j.adhoc.2015.08.023>
- [8] Tang LY, Li SN, Lin JH, Guo Q, Liu JG (2016). Community structure detection based on the neighbor node degree information. *International Journal of Modern Physics C*, pp. 1650046. <https://doi.org/10.1142/S0129183116500467>
- [9] Jiao Z, Ma K, Rong Y, Wang P, Zhang H, Wang S (2018). A path planning method using adaptive polymorphic ant colony algorithm for smart wheelchairs. *Journal of Computational Science*, pp. 50-57. <https://doi.org/10.1016/j.jocs.2018.02.004>
- [10] Zhou X, Liu Y, Zhang J, Liu T, Zhang D (2015) An ant colony based algorithm for overlapping community detection in complex networks. *Physica A Statistical Mechanics & Its Applications*, pp. 289-301. <https://doi.org/10.1016/j.physa.2015.02.020>
- [11] Chen S, Yang J, Li Y, Yang JF (2017). Multiconstrained Network Intensive Vehicle Routing Adaptive Ant Colony Algorithm in the Context of Neural Network Analysis. *Complexity*, pp. 1-9. <https://doi.org/10.1155/2017/8594792>
- [12] Ding J, He X, Yuan J, Chen Y, Jiang B (2018). Community detection by propagating the label of center. *Physica A Statistical Mechanics & Its Applications*, pp. S0378437118302632. <https://doi.org/10.1016/j.physa.2018.02.174>
- [13] Zhang C, Ni M, Yin H, Qiu K (2018). Developed Density Peak Clustering With Support Vector Data Description for Access Network Intrusion Detection. *IEEE Access*, pp. 46356-46362. <https://doi.org/10.1109/ACCESS.2018.2866128>
- [14] Du T, Qu S, Wang Q (2018) A Data-Driven Parameter Adaptive Clustering Algorithm Based on Density Peak. *Complexity*, pp. 1-14. <https://doi.org/10.1155/2018/5232543>
- [15] Allegra M, Seyed-Allaei S, Pizzagalli F, Baftizadeh F, Maieron M, Reverberi C, Laio A, Amati D (2017). fMRI single trial discovery of spatio-temporal brain activity patterns. *Human Brain Mapping*, pp. 1421-1437. <https://doi.org/10.1002/hbm.23463>

# Research on the Recognition Algorithm of Important Nodes in Complex Network

Yue Su

Guangzhou Huali Science and Technology Vocational College, Guangzhou 511325, China

E-mail: suyue\_ys@yeah.net

## Student paper

**Keywords:** complex network, important node, K-shell algorithm, node degree, zachary network

**Received:** December 26, 2019

*It is crucial to identify the important nodes in the network. On the basis of the K-shell algorithm, this study researched the recognition of important nodes in complex networks. The study introduces concepts of edge weight and influence coefficient, the design of an IKS algorithm, and an analysis of its recognition effect in Zachary network and real micro blog network. The results showed that the partition results of the K-shell algorithm were coarse, while the partition results of the IKS algorithm were refined; the IKS algorithm could sort the important nodes accurately on the basis of the K-shell algorithm, and its rationality was higher than that of the closeness centralization and PaperRank algorithm. The partition results in the microblog network also verified the effectiveness of the improved method. The experimental results show that the IKS algorithm is reliable in the important node identification, which is beneficial to the recognition of important nodes in complex network.*

*Povzetek: V študentskem prispevku je opisan študij algoritma za določitev pomembnih vozlišč v kompleksnih omrežjih.*

## 1 Introduction

Because of constant societal developments, people's daily activities are progressively closely linked. Most of these activities can be regarded as a kind of complex network. In the network, some areas are closely connected with each other, while the connection with other areas is very weak. These areas can be called the clustering property of the network. The network is composed of different communities, such as social network and information network, traffic network, etc. The number of members in complex networks is large, and individual behaviors and mutual relations are very complex [1]. With the development of technology, it is feasible to mine information in complex networks. The important nodes in the network community are the nodes that have an important impact on the network, which has very important practical significance for controlling rumor spread, the spread of infectious diseases [2], traffic conditions [3], etc. This is why the recognition algorithm of important nodes has been widely researched. Gu et al. [4] designed an important node identification algorithm based on LeadeRank algorithm and node similarity and verified the reliability of the algorithm by simulation on SIR model and Superman model. Based on the shortest path, Zheng et al. [5] combined proximity and centrality to find important nodes and found through performance analysis that the method was more effective in finding important nodes. Hu et al. [6] designed a multi-index method which integrated the centrality of feature vectors and degree centrality to recognize important nodes and found through simulation experiments that the algorithm was more reasonable and accurate. Wen et al. [7] designed

a "No Return" method for the importance evaluation of aviation network nodes and found that this method could effectively find potential important nodes with high accuracy after testing it on the aviation networks of China and the United States. In this study, the classical K-shell algorithm was improved and an IKS algorithm was designed for identifying the important nodes. Experiments on the Zachary and the Weibo networks verified the reliability of the method, which is conducive to effectively measuring the important nodes in complex networks and can be used in controlling the spread of infectious diseases and rumors.

## 2 Complex network

The complex network is usually described by the network graph. According to graph theory, the network is composed of node  $V$  and edge  $E$ , denoted as  $G = (V, E)$ , and the related concept includes:

- (1) degree. The number of edges of node  $v$  is called degree, represented by  $k$ . It is divided into out-degree  $k_i^{out}$  (number of edges which points at other nodes) and in-degree  $k_i^{in}$  (number of edges with other nodes pointing at the target node) according to the directions of edges.
- (2) average path length. Path with the least edges connecting node  $i$  and  $j$  is the shortest path, and the number of edges on the shortest path is the distance between the two nodes,  $d_{ij}$ . If the number of nodes is  $n$ , then average path length  $L$  can be expressed as:

$$L = \frac{1}{\frac{1}{2}n(n-1)} \sum_{i \geq j} d_{ij}$$

(3) clustering coefficient. For node  $i$ ,  $k_i$  nodes connect with it, then there are at most  $\frac{1}{2}k_i(k_i - 1)$  edges between them. If the number of edges is  $E_i$ , then the clustering coefficient can be expressed as  $C_i = \frac{2E_i}{k_i(k_i - 1)}$ , and the

average clustering coefficient is  $\bar{C} = \frac{1}{N} \sum_{i=1}^N C_i$ .

(4) diameter. The maximum value of distance between nodes in the network is  $D = \max_{i,j} d_{ij}$ .

### 3 Improved K-shell based important node recognition algorithm

#### 3.1 Classic K-shell algorithm

K-shell algorithm is a method proposed by Kitsak et al. [8]. The algorithm has low time complexity, a good practicability in a large complex network, and high recognition accuracy. The recognition steps are as follows:

- (1) the degree of all nodes in the network is calculated;
- (2) all nodes with degree of 1 are searched, and the nodes and their connecting edges are deleted. In time, nodes with a degree of 1 may reappear, and they are deleted as well until there is no node with a degree of 1 in the network. The deleted node forms 1-shell.
- (3) all nodes with degree of 2 are searched, and the deletion repeats until there is no node with degree of 2 in the network. The deleted nodes form 2-shell.
- (4) the above steps repeat until all nodes in the network have corresponding shell values.

Take Figure 1 as an example, (1) represents the original picture, (2) represents 1-shell, (3) represents 2-shell, and (4) represents 3-shell.

In K-shell algorithm, the larger the K-shell value of a node, the greater the influence on the network, and the lower the computational complexity of K-shell algorithm, which has a good division of the hierarchy. However, K-shell cannot show the differences of nodes in the same layer of network, and the results are coarse and not refined enough.

#### 3.2 Improved K-shell algorithm

In order to recognize the importance of nodes better, the K-shell algorithm is improved in this study. Firstly, concepts of weight of edge  $w_{ij}$  and influence coefficient  $e_{ij}$  are introduced:

(1)  $w_{ij} = k_i + k_j$ , where  $k_i$  and  $k_j$  represent the degree of node, and the weight is the sum of the degree of node. If two nodes have many neighbor nodes, it means that

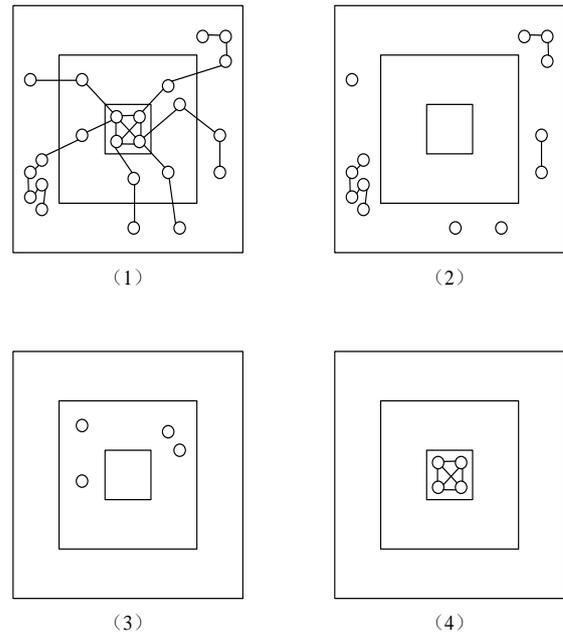


Figure 1: Examples of K-shell.

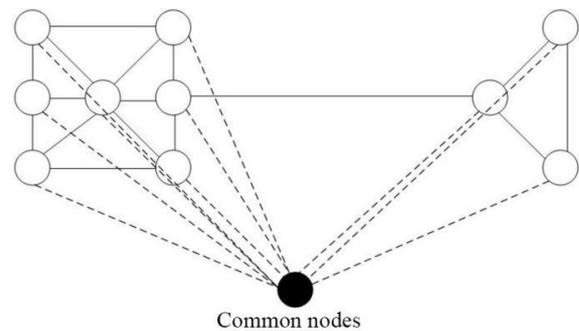


Figure 2: An example of common nodes.

more nodes will be involved if the information propagates in the two nodes.

(2)  $e_{ij} = \frac{|N_i \cap N_j|}{|N_i \cup N_j|}$ , where  $N_i$  and  $N_j$  represent the sets

of neighbor nodes, and the influence coefficient is the ratio of the number of common neighbors of two nodes to the total number of neighbors.

For the influence coefficient, if two nodes connect to two network communities but have no common friends, then  $e_{ij}$  of them is zero. In order to avoid this situation, common node is introduced (Figure 2), which increases the number of common neighbors and the total number of neighbors by 1 and avoids the influence coefficient to be zero.

Based on the above two concepts, weighted degree  $W_k$  of nodes is proposed:

$$W_{k_i} = \sum_{j \in N_i} e_{ij} k_{s_j} + \sum_{j \in N_i} w_{ij}$$

where  $k_{s_j}$  stands for the K-shell value of node  $j$ .

The improved K-shell algorithm is denoted as IKS algorithm, and its specific steps are shown in Figure 3.

In the IKS algorithm, the larger the IKS value, the larger the importance of the node, that is, the important node.

### 4 Example analysis

#### 4.1 Zachary network analysis

Zachary network is a common data set in network analysis, and has 34 nodes, including two communities with node 1 as the core and node 34 as the core, as shown in Figure 4.

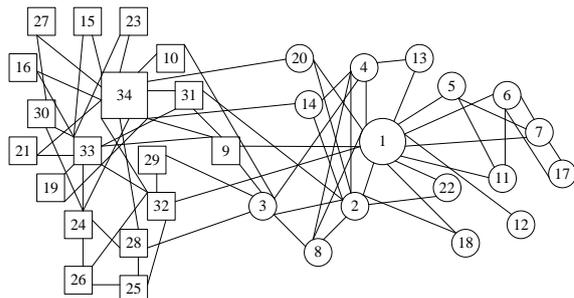


Figure 4: Zachary network.

The classical K-shell algorithm was used to identify the important nodes, and the results are shown in Table 1.

1-shell	12
2-shell	10, 13, 15, 16, 17, 18, 19, 21, 22, 23, 27
3-shell	5, 6, 7, 11, 20, 24, 25, 26, 28, 29, 30, 32
4-shell	1, 2, 3, 4, 8, 9, 14, 31, 33, 34

Table 1: Recognition results of the classic K-shell.

It was found from Table 1 that K-shell divides the whole network into four layers, among which 10 nodes in 4-shell could be regarded as important nodes, but the importance of these 10 nodes was further distinguished. It was found from Figure 4 that the importance of these 10 nodes was different, for example, node 1 was obviously more important than node 8.

In order to verify the reliability of the method, the Zachary network was identified by using closeness centralization [9], PaperRank [10] and IKS algorithm designed in this study. The top 10 nodes were taken as important nodes, and the results are shown in Table 2.

Rank	Closeness centralization	PaperRank	IKS
1	1	34	1
2	3	1	34
3	34	33	33
4	32	3	2
5	9	2	3
6	33	32	4
7	14	4	14
8	20	24	8
9	2	9	9
10	4	14	32

Table 2: Comparison of recognition results.

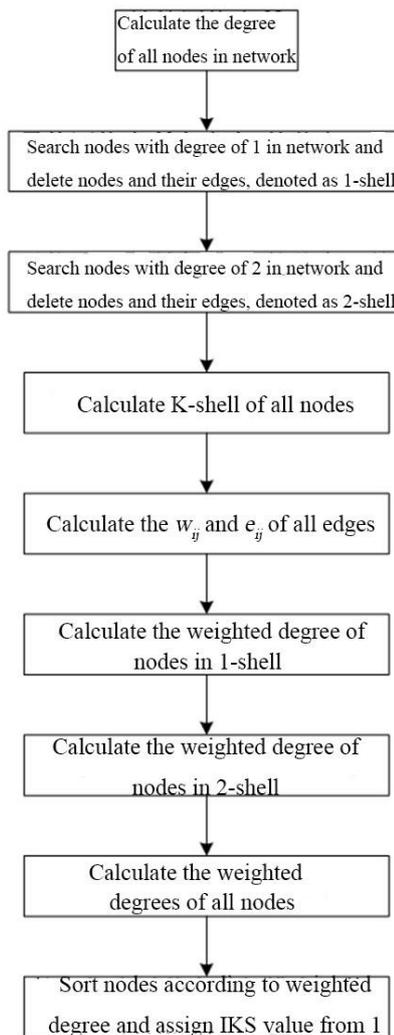


Figure 3: Steps of IKs algorithm.

It was found from Table 2 that the important nodes obtained by the three algorithms were basically similar, but there were some differences in the specific sorting. In proximity centrality, the importance of nodes was measured by the average distance from nodes to other nodes, and node 3 ranked the second place, while node 3 did not appear in the top three in PaperRank and IKS. Comparing node 3 with node 34, although node 3 was in the center of the network, node 34, as the core of a community, was significantly more important than node 3.

The sorting results of PaperRank and IKS were highly similar. Next, the differences were analyzed:

(1) node 34 and node 1. In the 4-shell, nodes 2, 3, 4, 8, 9 and 14 nodes connected with node 1, and nodes 9, 14, 31 and 33 connected with node 34. In the aspect of the number of neighbor nodes, node 1 was more important than node 34, which showed that the results of IKS algorithm were more reasonable.

(2) node 2 and node 3. It was found from Figure 2 that node 2 was more closely related to the community with node 1 as the core, while node 3 was related to both communities, but less closely than node 2, indicating that node 2 was more important than node 3.

(3) node 24 and node 8. Node 8 is not classified as the important node by PaperRank, and node 24 was also not classified as the important node by IKS. The K-shell division results showed that node 8 belonged to 4-shell and node 24 belonged to - 3 shell, indicating that node 8 was more important than node 24, and IKS division results were more accurate.

### 4.2 Microblog network analysis

Collecting micro blog data with web crawler obtained a data set with 4833 nodes, including 14256 edges, 13 network diameter, 1 minimum degree and 127 maximum degree, which is a complex network. Firstly, it was decomposed using K-shell algorithm, and the results are shown in Figure 5.

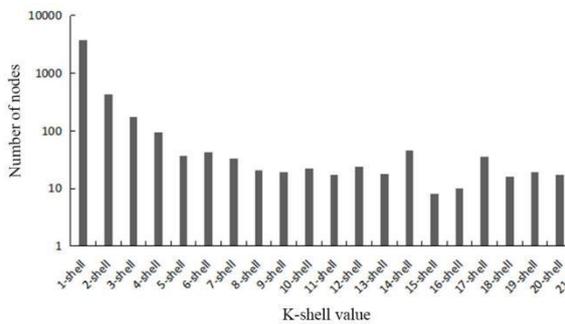


Table 3: Decomposition results of K-shell.

It was found from Figure 5 that the classic K-shell divided the whole network into 21 layers, of which the number of 1-shell nodes reached 3678 and the number of 21-shell nodes was 73. These 73 nodes could be regarded as important nodes, but the importance was not further distinguished. Therefore, the network was subdivided by the IKS algorithm, and the division results are shown in Table 3.

It was found from Table 3 that the IKS algorithm finally divided the network into 77 layers, the number of

IKS value	Number of nodes
1	1627
2	952
3	658
4	364
5	246
6	183
7	96
...	...
68	1
69	1
70	1
71	1
72	1
73	1
74	1
75	1
76	1
77	1

Table 4: Division results of IKS.

nodes with the minimum IKS value was 1627, which were the nodes with the smallest influence in the network, and the number of nodes with the maximum IKS value was 1, which was the most important node in the whole network. In the comparison of the division results between the classical K-shell algorithm and IKS algorithm, the number of nodes with low importance was large, while the number of nodes with high importance was small, which was consistent with the actual situation of microblog network. Compared with K-shell algorithm, IKS had a more precise recognition of important nodes, and the number of the top 10 important nodes was 1, which showed that IKS algorithm had strong recognition ability. It showed that the IKS algorithm made up for the defect of the rough division of the classic K-shell algorithm and could effectively sort the important nodes.

## 5 Discussion

Complex networks are pervasive in many fields, including biology [11], physics, computer science and others. Presently, the research encompasses important node identification [12], community discovery [13], link prediction [14], etc. Important node identification is a key problem in complex networks [15]. In any network, there are differences in the importance of nodes, where the important nodes play a key role, and it is of great significance to identify them [16]. For example, in the network of criminal gangs, through the analysis of the relationship between people, their leader can be localized and the police force can be centralized for control; in the network of infectious diseases, the source of infectious diseases can be found through the analysis of the network, so as to effectively isolate the source of diseases and slow down the spread of diseases; in the network of rumor propagation, the key figures can be mined; in the power network, the important nodes can be recognized and protected to effectively avoid large-scale failure [17]. Therefore, the identification of important nodes has a high practical value, and it is also of great significance to promote the development of related fields.

Currently, the commonly used recognition algorithms for important nodes are degree centrality (the larger the node degree, the more important it is), betweenness centrality (the more information the node propagates, the more important it is), closeness centralization (the more central the node is in the network, the more important it is), PaperRank algorithm (judging the importance of nodes according to the number and quality of other nodes pointing to the target node), etc. Based on the K-shell algorithm, an improved K-shell (IKS) algorithm was proposed in this study. On the basis of K-shell division, the importance of nodes was further sorted, so as to get more precise results. Additional experiments were carried out in Zachary network and a real micro blog network. It was found that the classical K-shell algorithm could identify important nodes, but the important nodes were not distinguished carefully, and the IKS algorithm could effectively improve the results of the K-shell algorithm and accurately identified the important nodes in the network. In the comparison with closeness centralization

and PaperRank algorithms, it was also found that the IKS algorithm designed in this study had higher reliability in the recognition of important nodes.

Although some achievements were made in the research of important nodes identification, there are still some shortcomings due to the limited time and ability:

(1) only the static network was analyzed, but the actual network has dynamic changes; the effectiveness of the algorithm on the dynamic network needs to be studied;

(2) whether the algorithm is equally applicable to the important node identification of weighted networks needs to be studied.

## 6 Conclusion

The recognition of important nodes in complex networks was studied in this work, the IKS algorithm was obtained by improving the classical K-shell algorithm, and experiments were carried out on the Zachary network and a micro blog network. The results showed that:

(1) the classical K-shell algorithm could divide important nodes, but it cannot sort them in details;

(2) compared with closeness centralization and PaperRank algorithm, the results of IKS were more reasonable;

(3) the IKS algorithm could effectively improve the coarse division results of the K-shell algorithm and realize the accurate identification of important nodes.

## 7 References

- [1] Tong C, Lian Y, Niu J, Xie Z, Zhang Y (2016). A novel green algorithm for sampling complex networks. *Journal of Network and Computer Applications*, pp. 55-62.  
<https://doi.org/10.1016/j.jnca.2015.05.021>
- [2] Jahanpour E, Chen X (2013). Analysis of complex network performance and heuristic node removal strategies. *Communications in Nonlinear Science and Numerical Simulation*, pp. 3458-3468.  
<https://doi.org/10.1016/j.cnsns.2013.04.030>
- [3] Wang HY, Wen RY, Zhao YF (2015). Empirical Research on Topological Characteristics of Air Traffic Situation Network. *Applied Mechanics and Materials*, pp. 1975-1979.  
<https://doi.org/10.4028/www.scientific.net/AMM.744-746.1975>
- [4] Gu YR, Zhu ZY (2017). Node Ranking in Complex Networks Based on LeaderRank and Modes Similarity. *Journal of the University of Electronic Science and Technology of China*, pp. 441-448.  
<https://doi.org/10.3969/j.issn.1001-0548.2017.02.020>
- [5] Zhang Z, Zhang ZY, Song MM (2013). Important node searching algorithm based on shortest-path betweenness. *Computer Engineering & Applications*, pp. 98-97.
- [6] Hu F, Liu Y (2015). Multi-index algorithm of identifying important nodes in complex networks based on linear discriminant analysis. *Modern Physics Letters B*, pp. 1450268.  
<https://doi.org/10.1142/s0217984914502686>
- [7] Wen X, Tu C, Wu M (2018). Node importance evaluation in aviation network based on “No Return” node deletion method. *Physica A: Statistical Mechanics and its Applications*, pp. S0378437118301997.  
<https://doi.org/10.1016/j.physa.2018.02.109>
- [8] Carmi S, Havlin S, Kirkpatrick S, Shavitt Y, Shir E (2007). From the Cover: A model of Internet topology using k-shell decomposition. *Proceedings of the National Academy of Science*, pp. 11150-11154.  
<https://doi.org/10.1073/pnas.0701175104>
- [9] Basu S, Maulik U, Chatterjee O (2016). Stability of Consensus Node Orderings Under Imperfect Network Data. *IEEE Transactions on Computational Social Systems*, pp. 1-12.  
<https://doi.org/10.1109/TCSS.2016.2596041>
- [10] Shuang X, Wang P, Zhang CX, Lu J (2018). Spectral Learning Algorithm Reveals Propagation Capability of Complex Networks. *IEEE Transactions on Cybernetics*, pp. 1-9.  
<https://doi.org/10.1109/TCYB.2018.2861568>
- [11] Saucède T, Laffont R, Labrière C, Jebrane A, François E, Eble GJ, David B (2015). Empirical and theoretical study of atelostomate (Echinoidea, Echinodermata) plate architecture: using graph analysis to reveal structural constraints. *Paleobiology*, pp. 436-459.  
<https://doi.org/10.1017/pab.2015.7>
- [12] Han ZM, Wu Y, Tan XS, Duan DG, Yang WJ (2015). Ranking key nodes in complex networks by considering structural holes. *Acta Physica Sinica*, pp. 58902-058902.  
<https://doi.org/10.7498/aps.64.058902>
- [13] Deng X, Wen Y, Chen Y (2016). Highly efficient epidemic spreading model based LPA threshold community detection method. *Neurocomputing*, pp. 3-12.  
<https://doi.org/10.1016/j.neucom.2015.10.142>
- [14] Li L, Qian L, Wang X, Luo S, Chen X (2015). Accurate similarity index based on activity and connectivity of node for link prediction. *International Journal of Modern Physics B*, pp. 1550108.  
<https://doi.org/10.1142/S0217979215501088>
- [15] Zhong LF, Shang MS, Chen XL, Cai SM (2018). Identifying the influential nodes via eigen-centrality from the differences and similarities of structure. *Physica A: Statistical Mechanics and its Applications*, pp. 77-82.  
<https://doi.org/10.1016/j.physa.2018.06.115>
- [16] Xu S, Wang P (2017). Identifying important nodes by adaptive LeaderRank. *Physica A: Statistical Mechanics and its Applications*, pp. 654-664.  
<https://doi.org/10.1016/j.physa.2016.11.034>
- [17] Li Y, Zhu W, Huang C, Xiong N (2017). Research on power heterogeneous communications network stability with SOC. *Power System Protection and Control*, pp. 118-122.  
<https://doi.org/10.7667/PSPC160342>



# Interactive Synthesis and Visualisation of Vast Areas with Geometrically Diverse Trees

Štefan Kohek

Faculty of Electrical Engineering and Computer Science, University of Maribor, Slovenia

E-mail: stefan.kohek@um.si

## Thesis summary

**Keywords:** generation of trees, particle flow simulation, GPU, volumetric rendering

**Received:** April 9, 2020

*This paper summarises a Doctoral Thesis which proposes a new approach for large-scale forest visualisation with geometrically diverse trees. The main contribution of the proposed method is an interactive visualisation of numerous trees without generating geometric data in advance, which is achieved by a new method for on-the-fly tree skeleton synthesis with a specific level of detail, and by a new procedural volumetric tree crown visualisation which avoids geometry formation altogether. The proposed method enables visualisation of forests with millions of trees, thus allowing rendering more trees than geometry-based visualisation methods.*

*Povzetek: Prispevek povzema doktorsko disertacijo, ki predlaga nov pristop za upodabljanje obsežnejših gozdov z geometrijsko raznolikimi drevesi. Glavni prispevek predlagane metode je interaktivno upodabljanje večjega števila dreves brez potrebe po vnaprej pripravljenih geometrijskih podatkih, kar dosežemo z novo metodo za sprotno tvorbo okostij dreves v določenem nivoju podrobnosti in z novo metodo za volumetrično upodabljanje krošenj dreves, ki se v celoti izogne tvorbi geometrijskih podatkov. Predlagana metoda omogoča upodabljanje gozdov z milijoni dreves in tako omogoča upodabljanje večjega števila dreves kot metode na podlagi vnaprej pripravljenih geometrijskih podatkov.*

## 1 Introduction

Trees are natural objects which consist of numerous leaves and branches. Various applications require convincing visualisation of vast areas with trees. In contrast to artificial objects, which consist mainly of flat surfaces, trees are geometrically more complex and require significant amounts of memory for geometric representation. Therefore, large-scale visualisation of geometrically diverse trees is challenging, due to memory constraints. Numerous techniques for forest rendering were developed in the past [1], which only partially address these issues through various techniques (e.g., instancing, pre-processing, or parallelisation).

This paper summarises a PhD thesis [2], which proposes a comprehensive pipeline for large-scale synthesis and visualisation of geometrically diverse trees. The main findings are published in the corresponding paper [3]. The following sections summarise the proposed approach, the main results and findings.

## 2 Overview of the proposed method

The main aim of the proposed approach is interactive visualisation of numerous geometrically diverse trees while navigating through the forest. In order to achieve convincing visualisation of nearby trees, the nearest trees need to

be visualised at the highest level of detail. In contrast, more distant trees can be visualised at lower levels of detail with lower memory requirements. However, in the case of millions of trees, geometric data of at least the most distant trees need to be omitted altogether due to memory limitations.

The trees in the proposed approach are generated procedurally by a new tree synthesis algorithm, based on a particle flow simulation, which generates tree skeletons of branches and leaves. The target tree structure is defined by a tree crown envelope and a few parameters, which define the branching pattern. The generated tree skeletons are visualised by generating actual geometric data directly on a Graphics Processing Unit (GPU).

In order to generate more distant trees without post-processing directly at lower levels of detail, multiple simplification schemes are introduced and integrated directly into the tree skeleton construction. However, tree synthesis needs to be fast, in order to achieve low latency of on-the-fly tree synthesis while navigating through the forest. Therefore, a new parallel tree synthesis algorithm for direct execution on the GPU is proposed, which is designed specifically for parallel tree synthesis of numerous trees.

To avoid geometry formation of the most distant trees completely, the thesis proposes a new procedural volumetric visualisation algorithm of tree crowns within the graph-

ics pipeline. Based on the defined tree crown envelopes, the volumetric visualisation method achieves identical appearance in comparison to the geometry-based visualisation. In this way, visual continuity is preserved when switching between the geometry-based and volumetric visualisation.

Finally, a new quad tree based framework is proposed, which integrates tree synthesis and visualisation algorithms to enable interactive visualisation and on-the-fly synthesis of the nearest trees while navigating through the forest.

### 3 Results

The proposed method was verified in terms of the tree synthesis duration, forest rendering rates, interactivity when navigating through forests, and similarity preservation of trees generated at lower levels of detail to the trees generated with the highest level of detail. Forests consisting of 400,000 branch segments were generated in less than 25 ms on the GPU Nvidia GTX 1060 with 4 TFLOPS of processing power, which enables generating larger numbers of trees between individual frames and preserving interactive rendering rates. Generating the trees with the lower levels of detail accelerated the synthesis even further, and enabled higher rendering rates. Additionally, the proposed simplification schemes generally achieved a gradual transition of similarity with the lower amount of memory. Hausdorff distance and precision and recall metrics, which were used for calculating similarity, agreed with these findings.

Performance of the geometry-based rendering was related closely to the amount of branch segments and leaves. In contrast, procedural volumetric rendering outperformed geometric rendering at a higher number of leaves or at a lower rendering resolution, which coincides with the visualisation of more distant trees. Overall, the proposed framework achieved stable rendering rates of more than 12 frames per second when displaying forests consisting of one million of trees.

### 4 Conclusion

The PhD thesis [2] has proposed a new comprehensive forest visualisation method, which enables detailed visualisation of close up trees and large-scale visualisation of distant trees without requiring the existing tree geometry. Procedural volumetric rendering is useful, especially for visualisation of distant trees, while a combination of tree skeleton synthesis and geometric rendering is used for the nearest trees.

### References

- [1] Smelik, R. M., Tutenel, T., Bidarra, R., and Benes, B. (2014). A Survey on Procedural Modelling for Virtual Worlds. *Computer Graphics Forum*, 33(6): 31–50. <https://doi.org/10.1111/cgf.12276>
- [2] Kohek, Š. (2019). *Interaktivna tvorba in prikaz obsežnih področij geometrijsko raznolikih dreves : doktorska disertacija*. PhD thesis, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko.
- [3] Kohek, Š., and Strnad, D. (2018). Interactive large-scale procedural forest construction and visualization based on particle flow simulation. *Computer Graphics Forum*, 37(1): 389–402. <https://doi.org/10.1111/cgf.13304>

## JOŽEF STEFAN INSTITUTE

*Jožef Stefan (1835-1893) was one of the most prominent physicists of the 19th century. Born to Slovene parents, he obtained his Ph.D. at Vienna University, where he was later Director of the Physics Institute, Vice-President of the Vienna Academy of Sciences and a member of several scientific institutions in Europe. Stefan explored many areas in hydrodynamics, optics, acoustics, electricity, magnetism and the kinetic theory of gases. Among other things, he originated the law that the total radiation from a black body is proportional to the 4th power of its absolute temperature, known as the Stefan–Boltzmann law.*

The Jožef Stefan Institute (JSI) is the leading independent scientific research institution in Slovenia, covering a broad spectrum of fundamental and applied research in the fields of physics, chemistry and biochemistry, electronics and information science, nuclear science technology, energy research and environmental science.

The Jožef Stefan Institute (JSI) is a research organisation for pure and applied research in the natural sciences and technology. Both are closely interconnected in research departments composed of different task teams. Emphasis in basic research is given to the development and education of young scientists, while applied research and development serve for the transfer of advanced knowledge, contributing to the development of the national economy and society in general.

At present the Institute, with a total of about 900 staff, has 700 researchers, about 250 of whom are postgraduates, around 500 of whom have doctorates (Ph.D.), and around 200 of whom have permanent professorships or temporary teaching assignments at the Universities.

In view of its activities and status, the JSI plays the role of a national institute, complementing the role of the universities and bridging the gap between basic science and applications.

Research at the JSI includes the following major fields: physics; chemistry; electronics, informatics and computer sciences; biochemistry; ecology; reactor technology; applied mathematics. Most of the activities are more or less closely connected to information sciences, in particular computer sciences, artificial intelligence, language and speech technologies, computer-aided design, computer architectures, biocybernetics and robotics, computer automation and control, professional electronics, digital communications and networks, and applied mathematics.

The Institute is located in Ljubljana, the capital of the independent state of Slovenia (or S<sup>o</sup>nia). The capital today is considered a crossroad between East, West and Mediter-

anean Europe, offering excellent productive capabilities and solid business opportunities, with strong international connections. Ljubljana is connected to important centers such as Prague, Budapest, Vienna, Zagreb, Milan, Rome, Monaco, Nice, Bern and Munich, all within a radius of 600 km.

From the Jožef Stefan Institute, the Technology park “Ljubljana” has been proposed as part of the national strategy for technological development to foster synergies between research and industry, to promote joint ventures between university bodies, research institutes and innovative industry, to act as an incubator for high-tech initiatives and to accelerate the development cycle of innovative products.

Part of the Institute was reorganized into several high-tech units supported by and connected within the Technology park at the Jožef Stefan Institute, established as the beginning of a regional Technology park “Ljubljana”. The project was developed at a particularly historical moment, characterized by the process of state reorganisation, privatisation and private initiative. The national Technology Park is a shareholding company hosting an independent venture-capital institution.

The promoters and operational entities of the project are the Republic of Slovenia, Ministry of Higher Education, Science and Technology and the Jožef Stefan Institute. The framework of the operation also includes the University of Ljubljana, the National Institute of Chemistry, the Institute for Electronics and Vacuum Technology and the Institute for Materials and Construction Research among others. In addition, the project is supported by the Ministry of the Economy, the National Chamber of Economy and the City of Ljubljana.

Jožef Stefan Institute  
Jamova 39, 1000 Ljubljana, Slovenia  
Tel.: +386 1 4773 900, Fax.: +386 1 251 93 85  
WWW: <http://www.ijs.si>  
E-mail: [matjaz.gams@ijs.si](mailto:matjaz.gams@ijs.si)  
Public relations: Polona Strnad

# INFORMATICA

## AN INTERNATIONAL JOURNAL OF COMPUTING AND INFORMATICS

### INVITATION, COOPERATION

#### Submissions and Refereeing

Please register as an author and submit a manuscript at: <http://www.informatica.si>. At least two referees outside the author's country will examine it, and they are invited to make as many remarks as possible from typing errors to global philosophical disagreements. The chosen editor will send the author the obtained reviews. If the paper is accepted, the editor will also send an email to the managing editor. The executive board will inform the author that the paper has been accepted, and the author will send the paper to the managing editor. The paper will be published within one year of receipt of email with the text in Informatica MS Word format or Informatica L<sup>A</sup>T<sub>E</sub>X format and figures in .eps format. Style and examples of papers can be obtained from <http://www.informatica.si>. Opinions, news, calls for conferences, calls for papers, etc. should be sent directly to the managing editor.

#### SUBSCRIPTION

Please, complete the order form and send it to Dr. Drago Torkar, Informatica, Institut Jožef Stefan, Jamova 39, 1000 Ljubljana, Slovenia. E-mail: [drago.torkar@ijs.si](mailto:drago.torkar@ijs.si)

Since 1977, Informatica has been a major Slovenian scientific journal of computing and informatics, including telecommunications, automation and other related areas. In its 16th year (more than twenty-six years ago) it became truly international, although it still remains connected to Central Europe. The basic aim of Informatica is to impose intellectual values (science, engineering) in a distributed organisation.

Informatica is a journal primarily covering intelligent systems in the European computer science, informatics and cognitive community; scientific and educational as well as technical, commercial and industrial. Its basic aim is to enhance communications between different European structures on the basis of equal rights and international refereeing. It publishes scientific papers accepted by at least two referees outside the author's country. In addition, it contains information about conferences, opinions, critical examinations of existing publications and news. Finally, major practical achievements and innovations in the computer and information industry are presented through commercial publications as well as through independent evaluations.

Editing and refereeing are distributed. Each editor can conduct the refereeing process by appointing two new referees or referees from the Board of Referees or Editorial Board. Referees should not be from the author's country. If new referees are appointed, their names will appear in the Refereeing Board.

Informatica web edition is free of charge and accessible at <http://www.informatica.si>.

Informatica print edition is free of charge for major scientific, educational and governmental institutions. Others should subscribe.

## Informatica WWW:

<http://www.informatica.si/>

### Referees from 2008 on:

A. Abraham, S. Abraham, R. Accornero, A. Adhikari, R. Ahmad, G. Alvarez, N. Anciaux, R. Arora, I. Awan, J. Azimi, C. Badica, Z. Balogh, S. Banerjee, G. Barbier, A. Baruzzo, B. Batagelj, T. Beaubouef, N. Beaulieu, M. ter Beek, P. Bellavista, K. Bilal, S. Bishop, J. Bodlaj, M. Bohanec, D. Bolme, Z. Bonikowski, B. Bošković, M. Botta, P. Brazdil, J. Brest, J. Brichau, A. Brodnik, D. Brown, I. Bruha, M. Bruynooghe, W. Buntine, D.D. Burdescu, J. Buys, X. Cai, Y. Cai, J.C. Cano, T. Cao, J.-V. Capella-Hernández, N. Carver, M. Cavazza, R. Ceylan, A. Chebotko, I. Chekalov, J. Chen, L.-M. Cheng, G. Chiola, Y.-C. Chiou, I. Chorbev, S.R. Choudhary, S.S.M. Chow, K.R. Chowdhury, V. Christlein, W. Chu, L. Chung, M. Cigliarić, J.-N. Colin, V. Cortellessa, J. Cui, P. Cui, Z. Cui, D. Cutting, A. Cuzzocrea, V. Cvjetkovic, J. Cyprianski, L. Čehovin, D. Čerepnalkoski, I. Čosić, G. Daniele, G. Danoy, M. Dash, S. Datt, A. Datta, M.-Y. Day, F. Debili, C.J. Debono, J. Dedič, P. Degano, A. Dekdouk, H. Demirel, B. Demoen, S. Dendamrongvit, T. Deng, A. Derezsinska, J. Dezert, G. Dias, I. Dimitrovski, S. Dobrišek, Q. Dou, J. Doumen, E. Dovgan, B. Dragovich, D. Dragic, O. Drbohlav, M. Drole, J. Dujmović, O. Ebers, J. Eder, S. Elaluf-Calderwood, E. Engström, U. riza Erturk, A. Farago, C. Fei, L. Feng, Y.X. Feng, B. Filipič, I. Fister, I. Fister Jr., D. Fišer, A. Flores, V.A. Fomichov, S. Forli, A. Freitas, J. Fridrich, S. Friedman, C. Fu, X. Fu, T. Fujimoto, G. Fung, S. Gabrielli, D. Galindo, A. Gambarara, M. Gams, M. Ganzha, J. Garbajosa, R. Gennari, G. Georgeson, N. Gligorić, S. Goel, G.H. Gonnet, D.S. Goodsell, S. Gordillo, J. Gore, M. Grčar, M. Grgurović, D. Grosse, Z.-H. Guan, D. Gubiani, M. Guid, C. Guo, B. Gupta, M. Gusev, M. Hahsler, Z. Haiping, A. Hameed, C. Hamzaçebi, Q.-L. Han, H. Hanping, T. Härder, J.N. Hatzopoulos, S. Hazelhurst, K. Hempstalk, J.M.G. Hidalgo, J. Hodgson, M. Holbl, M.P. Hong, G. Howells, M. Hu, J. Hyvärinen, D. Ienco, B. Ionescu, R. Irfan, N. Jaisankar, D. Jakobović, K. Jassem, I. Jawhar, Y. Jia, T. Jin, I. Jureta, Đ. Juričić, S. K, S. Kalajdziski, Y. Kalantidis, B. Kaluža, D. Kanellopoulos, R. Kapoor, D. Karapetyan, A. Kassler, D.S. Katz, A. Kaveh, S.U. Khan, M. Khattak, V. Khomenko, E.S. Khorasani, I. Kitanovski, D. Kocev, J. Kocijan, J. Kollár, A. Kontostathis, P. Korošec, A. Koschmider, D. Košir, J. Kovač, A. Krajnc, M. Krevs, J. Krogstie, P. Krsek, M. Kubat, M. Kukar, A. Kulis, A.P.S. Kumar, H. Kwašnicka, W.K. Lai, C.-S. Lai, K.-Y. Lam, N. Landwehr, J. Lanir, A. Lavrov, M. Layouni, G. Leban, A. Lee, Y.-C. Lee, U. Legat, A. Leonardis, G. Li, G.-Z. Li, J. Li, X. Li, X. Li, Y. Li, Y. Li, S. Lian, L. Liao, C. Lim, J.-C. Lin, H. Liu, J. Liu, P. Liu, X. Liu, X. Liu, F. Logist, S. Loskovska, H. Lu, Z. Lu, X. Luo, M. Luštrek, I.V. Lyustig, S.A. Madani, M. Mahoney, S.U.R. Malik, Y. Marinakis, D. Marinčič, J. Marques-Silva, A. Martin, D. Marwede, M. Matijašević, T. Matsui, L. McMillan, A. McPherson, A. McPherson, Z. Meng, M.C. Mihaescu, V. Milea, N. Min-Allah, E. Minisci, V. Mišić, A.-H. Mogos, P. Mohapatra, D.D. Monica, A. Montanari, A. Moroni, J. Mosegaard, M. Moškon, L. de M. Mourelle, H. Moustafa, M. Možina, M. Mrak, Y. Mu, J. Mula, D. Nagamalai, M. Di Natale, A. Navarra, P. Navrat, N. Nedjah, R. Nejabat, W. Ng, Z. Ni, E.S. Nielsen, O. Nouali, F. Novak, B. Novikov, P. Nurmi, D. Obrul, B. Oliboni, X. Pan, M. Pančur, W. Pang, G. Papa, M. Paprzycki, M. Paralič, B.-K. Park, P. Patel, T.B. Pedersen, Z. Peng, R.G. Pensa, J. Perš, D. Petcu, B. Petelin, M. Petkovšek, D. Pevec, M. Pičulin, R. Piltaver, E. Pirogova, V. Podpečan, M. Polo, V. Pomponiu, E. Popescu, D. Poshyvanik, B. Potočnik, R.J. Povinelli, S.R.M. Prasanna, K. Pripuzi, G. Puppis, H. Qian, Y. Qian, L. Qiao, C. Qin, J. Que, J.-J. Quisquater, C. Rafe, S. Rahimi, V. Rajković, D. Raković, J. Ramaekers, J. Ramon, R. Ravnik, Y. Reddy, W. Reimche, H. Rezankova, D. Rispoli, B. Ristevski, B. Robič, J.A. Rodriguez-Aguilar, P. Rohatgi, W. Rossak, I. Rožanc, J. Rupnik, S.B. Sadek, K. Saeed, M. Saeki, K.S.M. Sahari, C. Sakharwade, E. Sakkopoulos, P. Sala, M.H. Samadzadeh, J.S. Sandhu, P. Scaglioso, V. Schau, W. Schempp, J. Seberry, A. Senanayake, M. Senobari, T.C. Seong, S. Shamala, c. shi, Z. Shi, L. Shiguo, N. Shilov, Z.-E.H. Slimane, F. Smith, H. Sneed, P. Sokolowski, T. Song, A. Soppera, A. Sornioti, M. Stajdohar, L. Stanescu, D. Strnad, X. Sun, L. Šajn, R. Šenkeřík, M.R. Šikonja, J. Šilc, I. Škrjanc, T. Štajner, B. Šter, V. Štruc, H. Takizawa, C. Talcott, N. Tomasev, D. Torkar, S. Torrente, M. Trampuš, C. Tranoris, K. Trojancanec, M. Tschierschke, F. De Turck, J. Twycross, N. Tziritas, W. Vanhoof, P. Vateekul, L.A. Vese, A. Visconti, B. Vlaovič, V. Vojisavljević, M. Vozalis, P. Vračar, V. Vranić, C.-H. Wang, H. Wang, H. Wang, H. Wang, S. Wang, X.-F. Wang, X. Wang, Y. Wang, A. Wasilewska, S. Wenzel, V. Wickramasinghe, J. Wong, S. Wrobel, K. Wrona, B. Wu, L. Xiang, Y. Xiang, D. Xiao, F. Xie, L. Xie, Z. Xing, H. Yang, X. Yang, N.Y. Yen, C. Yong-Sheng, J.J. You, G. Yu, X. Zabulis, A. Zainal, A. Zamuda, M. Zand, Z. Zhang, Z. Zhao, D. Zheng, J. Zheng, X. Zheng, Z.-H. Zhou, F. Zhuang, A. Zimmermann, M.J. Zuo, B. Zupan, M. Zuqiang, B. Žalik, J. Žižka,

# *Informatica*

## An International Journal of Computing and Informatics

Web edition of Informatica may be accessed at: <http://www.informatica.si>.

**Subscription Information** Informatica (ISSN 0350-5596) is published four times a year in Spring, Summer, Autumn, and Winter (4 issues per year) by the Slovene Society Informatika, Litostrojska cesta 54, 1000 Ljubljana, Slovenia.

The subscription rate for 2020 (Volume 44) is

- 60 EUR for institutions,
- 30 EUR for individuals, and
- 15 EUR for students

Claims for missing issues will be honored free of charge within six months after the publication date of the issue.

Typesetting: Borut Žnidar, [borut.znidar@gmail.com](mailto:borut.znidar@gmail.com).

Printing: ABO grafika d.o.o., Ob železnici 16, 1000 Ljubljana.

Orders may be placed by email ([drago.torkar@ijs.si](mailto:drago.torkar@ijs.si)), telephone (+386 1 477 3900) or fax (+386 1 251 93 85). The payment should be made to our bank account no.: 02083-0013014662 at NLB d.d., 1520 Ljubljana, Trg republike 2, Slovenija, IBAN no.: SI56020830013014662, SWIFT Code: LJBASI2X.

Informatica is published by Slovene Society Informatika (president Niko Schlamberger) in cooperation with the following societies (and contact persons):

Slovene Society for Pattern Recognition (Vitimir Štruc)

Slovenian Artificial Intelligence Society (Sašo Džeroski)

Cognitive Science Society (Olga Markič)

Slovenian Society of Mathematicians, Physicists and Astronomers (Dragan Mihailović)

Automatic Control Society of Slovenia (Giovanni Godena)

Slovenian Association of Technical and Natural Sciences / Engineering Academy of Slovenia (Mark Pleško)

ACM Slovenia (Nikolaj Zimic)

Informatica is financially supported by the Slovenian research agency from the Call for co-financing of scientific periodical publications.

Informatica is surveyed by: ACM Digital Library, Citeseer, COBISS, Compendex, Computer & Information Systems Abstracts, Computer Database, Computer Science Index, Current Mathematical Publications, DBLP Computer Science Bibliography, Directory of Open Access Journals, InfoTrac OneFile, Inspec, Linguistic and Language Behaviour Abstracts, Mathematical Reviews, MatSciNet, MatSci on SilverPlatter, Scopus, Zentralblatt Math
---

# *Informatica*

**An International Journal of Computing and Informatics**

Improvement of the Deep Forest Classifier by a Set of Neural Networks	L.V. Utkin	<b>1</b>
Creation of Facial Composites from User Selections using Image Gradient	R. García-Zurdo	<b>15</b>
Design Optimization Average-Based Algorithm	A. Barreiros, J.B. Cardoso	<b>23</b>
The Iris Dataset Revisited. A Partial Ordering Study	L. Carlsen	<b>35</b>
Evaluating Websites of Conservation Labs in Museums using Fuzzy Multi-Criteria Decision Making Theories	K. Kabassi, A. Botonis, C. Karydis	<b>45</b>
Application of Algorithms with Variable Greedy Heuristics for k-Medoids Problems	L.A. Kazakovtsev	<b>55</b>
Hybrid Nearest Neighbors Ant Colony Optimization for Clustering Social Media Comments	L. Lucky, A.S. Girsang	<b>63</b>
A Robust Image Watermarking Scheme Based on the Laplacian Pyramid Transform	S.C. Nguyen, K.H. Ha, H.M. Nguyen	<b>75</b>
Feature Level Fusion of Face and Voice Biometrics Systems Using Artificial Neural Network for Personal Recognition	C. Dalila	<b>85</b>
Comparison of Community Structure Partition Optimization of Complex Networks by Different Community Discovery Algorithms	R. Peng, Y. Yao	<b>97</b>
Research on Recognition Algorithm of Important Nodes in Complex Network	Y. Su	<b>103</b>
Interactive Synthesis and Visualisation of Vast Areas with Geometrically Diverse Trees	Š. Kohek	<b>109</b>

