

Multi-Objective Particle Swarm Optimization-based Feature Selection for Face Recognition

Souad LARABI-MARIE-SAINTE^{1*}, Sanaa GHOUZALI²

¹ Computer Science Department, Prince Sultan University Riyadh, 11495, Saudi Arabia
slarabi@psu.edu.sa (*Corresponding author)

² Information Technology Department, King Saud University Riyadh, 11495, Saudi Arabia
sghouzali@ksu.edu.sa

Abstract: The curse of dimensionality is a well-known problem in biometric applications (e.g., biometric passports). The downside of this problem is that both the accuracy and speed of the biometric authentication process are reduced. This paper sets forth a feature selection (FS) method based on speed-constrained multi-objective particle swarm optimization (SMPSO). The proposed approach aims to reduce the size of the biometric features through the minimization of the intra-class variations and the maximization of the inter-class variations. Experiments have been conducted using several datasets from University of California–Irvine (UCI) to confirm the efficiency of SMPSO-based FS against state-of-the-art multi-objective FS approaches, such as the multi-objective evolutionary algorithm based on decomposition (MOEA/D) and the second non-dominated sorting genetic algorithm (NSGA-II). When compared to NSGA-II and MOEA/D, SMPSO gained 6.01% and 6.11%, respectively, in average classification accuracy. Moreover, SMPSO achieved the best accuracy compared to MOGA, a modified version of NSGA-II. The experimental results obtained by using a YALE Face database validated the effectiveness of the proposed approach in reducing the size of the biometric features while allowing a good recognition accuracy. The classification performance was improved by 8.2% compared with the performance of the state-of-the-art approaches.

Keywords: Feature selection, Particle swarm optimization, Multi-objective optimization, Face recognition.

1. Introduction

Information security is an essential field nowadays. Its fundamental services include authentication, which can be efficiently performed using biometric attributes related to physiological, cognitive, and/or behavioral characteristics (e.g., face and gait). An important issue of biometric authentication systems is the large dimensionality of a dataset, known as the “curse of dimensionality” problem, which can considerably increase the computational cost. Additionally, the authentication accuracy can decrease because of the restricted capabilities of the classifier (Jain, Duin & Mao, 2000). Therefore, feature selection (FS) has to be performed before the classification process. FS determines the most relevant features in a dataset, which increases the classification accuracy as compared to using all the features of large dimensional datasets.

The best-known dimensionality reduction approaches applied to face recognition that are surveyed by (He et al. 2005; Cai et al. 2006; Jafri & Arabnia, 2009) are principal component analysis (PCA), independent component analysis (ICA), linear discriminant analysis (LDA), locality preserving projection (LPP), and orthogonal LPP (OLPP). However, these methods cannot minimize the inter-class variation (i.e., the distance between templates from the same user) and maximize the intra-class variation (i.e., the distance between

different users) simultaneously. Moreover, the linearity of these methods represents another major disadvantage. To overcome this drawback, kernel-based methods can be applied to transform a nonlinear problem (in the original space) into a linear one, which is defined in the feature space. Henceforth, PCA, LDA, or LPP can consequently be executed in the feature space, which results in kernel principal component analysis (KPCA), kernel Fisher discriminant analysis (KFD), and kernel locality preservation projection (KLPP), respectively. Several studies have shown that KPCA, KFD, and KLPP can have better performance in applications like face recognition (Yang et al., 2005; Cheng et al., 2005). However, these methods can successfully reduce the feature dimensionality, but not necessarily improve the classification performance since the problem is solved as a single objective of identifying the relevant features. Thus, FS methods should at least consider two crucial objectives for maximizing the classification performance and minimizing the number of features simultaneously. In literature, optimization algorithms based on biological evolution, such as the methods focused on POA (population-based optimization algorithm), have successfully solved complex optimization problems. Since these POA methods can capture multiple optimal solutions,

they are appropriate for solving multi-objective optimization (MOO) problems by finding the best features that allow efficient classification of samples while reducing the dimensionality of the feature vectors. Moreover, POA-based MOO is more efficient for complex optimization problems that require substantial datasets, which include many complicated features. This type of methods comprises evolutionary algorithms (e.g., genetic algorithms (GA)) and swarm intelligence algorithms (e.g., particle swarm optimization (PSO), artificial bee colony). Recently, (Dinu, 2015) has successfully applied Multi-objective PSO to solve the problem of assembly line balancing for industrial applications. Besides, numerous researchers have demonstrated that PSO is an effective method for feature selection (FS) (e.g., Xue, Zhang & Browne, 2013). However, its use in multi-objective FS has not been explored enough, and only a few studies have been conducted in this context (Hamdani et al. 2007). Consequently, in this article, the speed-constrained multi-objective PSO (SMPSO) technique (Nebro et al., 2009) is analyzed in order to obtain relevant features of users' biometric templates by minimizing their intra-class variations and maximizing their inter-class variations.

The structure of the remainder of this paper is given as follows. Section 2 provides a brief literature review of bio-inspired based multi-objective FS techniques. The proposed approach is presented in Section 3. The obtained experimental results are discussed in Section 4. Finally, Section 5 includes the conclusion and recommendations for future work.

2. State-of-the-Art

In this section, the main bio-inspired based multi-objective FS approaches are discussed, including Elicit Non-dominated Sorting Genetic algorithm (NSGA-II), Multi-objective Evolutionary Algorithms (MOEA), and Multi-objective Particle Swarm Optimization (MOPSO).

NSGA-II has been remarkably applied to solve the FS problem. (Hamdani et al. 2007) evaluated the performance of NSGA-II using a 1-nearest neighbor (NN) classifier to minimize the number of selected features and the classification error for five datasets from the University of California-Irvine (UCI). The obtained results are found

to be promising. (Ekbal, Saha & Garbe, 2010) handled the problem of named entity recognition using the maximum entropy classifier with two different languages. They optimized recall and precision classification quality measures. The proposed method yielded promising results. The previously mentioned work was extended by (Ekbal & Saha, 2012) by using three different classifiers (maximum entropy, conditional random field, and support vector machines) and three languages. Their obtained results illustrated an improvement in recall and precision. (Venkatadri & Srinivasa, 2010) proposed a modified version of NSGA-II, called MOGA to handle FS. Different pairs of objective functions were optimized, corresponding to different measures that identified relevant features (inter-class distance and attribute class correlation, inter-class distance and Laplacian score, inconsistent pairs, and Laplacian score). MOGA was tested with UCI datasets using a decision tree classifier. The experimental results showed relatively higher accuracy rates for all datasets when using the inter-class distance measure. One suggested that further investigation regarding this criterion should be carried out, such as combining three objectives simultaneously, including the number of features. Another study by (Soyel, Tekguc & Demirel, 2011) employed NSGA-II for facial expression recognition using two objective functions, including the number of features and the Fisher criterion. The proposed technique was validated through a 3D facial expression database (i.e., BU-3DFE) using probabilistic neural networks. The results enhanced facial expression recognition performance. In (De-la-Hoz et al., 2014), NSGA-II was employed to address the network anomaly detection problem. The hierarchical self-organizing map classifier was utilized. The proposed method outperformed the existing FS methods. (Tan, Lim et Cheah, 2014) proposed a modified micro-GA based FS. A voting-based elite selection classifier was used as well as three objective functions, namely the specificity rate, the sensitivity rate, and the number of selected features. This method was tested for two UCI datasets and successfully compared with several state-of-the-art FS methods.

MOEA has also been successfully applied to solve the FS problem. (Paul & Das, 2015) optimized the inter-class and intra-class distances using a k-nearest neighbor (k-NN) classifier.

The results obtained for UCI datasets were successfully compared to those obtained by employing different existing single- and multi-objective FS methods.

MOPSO was first employed for the FS concept by (Xue, Zhang & Browne, 2013). Two schemes were proposed: one based on non-dominated sorting, and the other one employing crowding, mutation, and dominance, to minimize the number of features and the classification error rate. The proposed approaches were compared with single- and multi-objective FS methods. The experiments by using 12 UCI datasets showed that the implementation of the first scheme provided results which might be compared with those obtained by using other algorithms, whereas the second provided the best results among all other algorithms. (Annavaarapu, Dara & Banka, 2016) proposed another MOPSO based FS study that used two objective functions (the number of features and the capability of the designated features). The proposed technique was compared to NSGA-II and GA using three well-known gene expression datasets (i.e., Colon, Leukemia, and Lymphoma). The experiment results were encouraging. (Zhang, Gong & Cheng, 2017) applied MOPSO to minimize the cost and the number of feature subsets. The proposed technique was successfully compared to existing techniques using five benchmark datasets.

To sum up, most of the FS methods proposed in the specialized literature were based on NSGA-II. Moreover, NSGA-II was successfully employed for facial expression recognition by (Soyel, Tekguc & Demirel, 2011). On the other hand, only three studies investigated the use of MOPSO to solve the FS problem. Therefore, it is of interest to further investigate the use of a new FS method applying MOPSO for face recognition.

3. Proposed Approach

To solve the face recognition problem, the classes of individuals should be determined such that the intra-class variations are minimized, and the inter-class variations are maximized. This paper proposes an approach which concurrently optimizes two different measures of classification using the searchability of a Multi-Objective Optimization algorithm (Mandal, Mukhopadhyay & Dutta, 2018; Yann & Siarry, 2013). The primary purpose of MOO is to create a Pareto front of non-

dominated solutions (feature subsets). A decision vector u is said to strictly dominate a vector v if $f_i(u) \leq f_i(v) \forall i = 1, \dots, K$ (K is the number of objective functions) and $f_i(u) < f_i(v)$ for some i . A set of non-dominated solutions requires that no member of that set be dominated by any other member. The representation of the Pareto optimal solutions in the objective space constitutes the Pareto front. Figure 1 shows an example of a Pareto front for the problem of optimizing two objectives (inter-class distance and intra-class distance). The black points represent the Pareto front (the non-dominated solutions). In this study, the non-dominated solutions are m -dimensional vectors, where m is the number of features. The solutions' values range between 0 and 10. Let us consider $u = (u_1, u_2, \dots, u_m)$ a non-dominated solution. If $u_1 = 0$ then, the feature i ($i = 1, \dots, m$) is removed otherwise, it is kept. The aforementioned concept is well explained in Section 3.2.

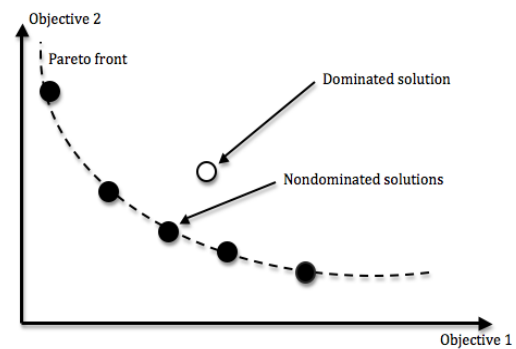


Figure 1. Example of a Pareto Front. © 2020 IEEE. Reprinted, with permission, from (Larabi-Marie-Sainte, 2015)

For further details concerning the principle concept of MOO, one should consult (Mandal, Mukhopadhyay & Dutta, 2018; Yann & Siarry, 2013). Commonly, obtaining the analytical expression of Pareto front (called True Pareto front) is not possible, however, three main goals should be satisfied: 1) increasing the number of non-dominated solutions, 2) approaching the Pareto front generated by the proposed algorithm to the True Pareto front if its location is identified, and 3) making the distribution of the non-dominated solutions uniform.

Different existing MOO algorithms use two critical notions that need to be discussed: crowding distance and archive.

Crowding distance: is a density measure whose calculation, for a solution j , is based upon the points which are near to j for each objective function, as described in Algorithm 1 (Sheikholeslami & Navimipour, 2017).

Archive: is an external memory employed for saving the non-dominated solutions obtained throughout the procedure. Each candidate included in the archive must be non-dominated or dominate the archive members. Dominated solutions in the archive must be disconnected. This characterization guarantees that the archive members are all non-dominated after each iteration.

Algorithm 1. Crowding Distance (Sheikholeslami & Navimipour, 2017)

Input: K : The number of objectives;
 A : set of solutions for which the crowding distance is calculated;
 $sort(A, m)$: Sort the solutions found in A according to the m^{th} objective in ascending order;
 $l = |A|$: The cardinal of the set of solutions A
*/*Initialize the distance*/*
For $i \leftarrow 1$ **to** l **do**
 $A[i]_{distance} \leftarrow 0$;
End
For $m \leftarrow 1$ **to** K **do**
 / Sort using each objective value*/*
 $I \leftarrow sort(A, m)$;
 / The distance of the boundary points are fixed to infinity*/*
 $A[1]_{distance} \leftarrow \infty$;
 $A[l]_{distance} \leftarrow \infty$;
 For $i \leftarrow 1$ **to** $l - 1$ **do**
 $A[i]_{distance} \leftarrow A[i]_{distance} + \frac{f_{i+1}^m - f_{i-1}^m}{f_{i+1}^{Max} - f_{i-1}^{Min}}$;
 End
End

3.1 Objective Functions

Before addressing the objective functions, it is necessary to introduce the normalization notion. Consider a dataset matrix Y of size $n * m$, where m is the number of features and n is the number of data points. Then each y_{ij} denotes the i^{th} data point of the j^{th} feature. Since the values of a given dataset can have different orders, normalization was applied for the interval of $[0, 10]$ as recommended by (Paul & Das, 2015).

The aim of this study is to determine the classes of individuals by providing:

- **Objective 1:** the minimization of the variance within the same group so that it is homogeneous (Equation 1) and;
- **Objective 2:** the maximization of the variance between the groups so that they are well separated (Equation 2).

$$D_{intra} = \sum_{p=1}^n \sum_{q=p+1}^n W^T |X_p - X_q| \quad \forall p, q \in \text{same class} \quad (1)$$

$$D_{inter} = \sum_{p=1}^n \sum_{q=p+1}^n W^T |X_p - X_q| \quad \forall p, q \notin \text{same class} \quad (2)$$

where X_p and X_q are the p^{th} and q^{th} rows of the normalized dataset matrix X ($p, q = 1, n$) and W is the weighting vector just like w_i is the weighting factor for the feature i , defined as follows:

$$w_i = \begin{cases} 0 & \text{if the feature is rejected} \\ [0, 10] & \text{if the feature is selected} \end{cases}$$

3.2 Speed-constrained Multi-Objective Particle Swarm Optimization (SMPSO)

SMPSO, as proposed by (Nebro et al., 2009), is investigated in this paper for the first time to solve multi-objective FS for face recognition. The SMPSO algorithm uses the concepts of dominance, mutation, and crowding to seek out non-dominated solutions. This technique has been successfully compared with NSGA-II and other well-known algorithms, and it has been characterized by its fast convergence towards the Pareto front and uniform distribution of non-dominated solutions (Durillo et al., 2009). Moreover, this technique has shown effectiveness in projection pursuit for detecting clusters in high dimensional datasets (Larabi-Marie-Sainte, 2011).

SMPSO follows the concept of PSO. A population of particles (i.e., swarm) is randomly generated with velocities in $[0, 10]$. At each iteration, each particle's position is updated based on its best-achieved position ($pbest$), the best particle of the swarm ($gbest$), and its velocity, as indicated in Equation 3.

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \quad (3)$$

where $\vec{x}_i(t)$ is the position of the particle i at the iteration t and $\vec{v}_i(t)$ is the velocity of particle i at iteration t . A particle's position is kept inside the search space by using Equation 4.

$$x_{ij} = \begin{cases} 10 & \text{if } x_{ij} > 10 \\ 0 & \text{if } x_{ij} < 0 \\ x_{ij} & \text{otherwise} \end{cases} \quad (4)$$

The neighborhood principle is not applied. The velocity is defined as:

$$\vec{v}_i(t) = w\vec{v}_i(t-1) + c_1r_1(\vec{x}_{pbest_i} - \vec{x}_i) + c_2r_2(\vec{x}_{gbest} - \vec{x}_i) \quad (5)$$

where $w \in [0.1, 0.5]$ is the inertia weight of the particle. The parameters $r_1, r_2 = \text{random}(0,1)$ and $c_1, c_2 \in [1.5, 2.5]$ control the global and personal best particles.

To master the velocity of a particle, the accumulated velocity can be obtained by multiplying the velocity (Equation 5) by χ , a constriction coefficient introduced by (Clerc & Kennedy, 2002).

$$\vec{v}_i(t) = \chi \vec{v}_i(t) \quad (6)$$

$$\chi = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}} \quad (7)$$

where

$$\varphi = \begin{cases} c_1 + c_2 & \text{if } c_1 + c_2 > 4 \\ 1 & \text{otherwise} \end{cases} \quad (8)$$

Such that:

$$v_{ij}(t) = \begin{cases} \delta_j & \text{if } v_{ij}(t) > \delta_j \\ -\delta_j & \text{if } v_{ij}(t) \leq -\delta_j \\ v_{ij}(t) & \text{otherwise} \end{cases} \quad (9)$$

$$\delta_j = \frac{\text{upper_limit}_j - \text{lower_limit}_j}{2} \quad (10)$$

$$\text{upper_limit}_j = 10 \text{ and } \text{lower_limit}_j = 0$$

Each particle is evaluated based on both objective functions (inter- and intra-class distances), and then the particle's *pbest* (memory) and the archive are updated. The choice of the *gbest* particle is made by picking up two solutions from the archive randomly and selecting the one with the largest crowding distance. The update of the *pbest* particle is performed by comparing the current position of the particle and its *pbest* (memory). *pbest* equals the current position if the latter dominates *pbest* or if both *pbest* and the current position are non-dominated. The archive will then be updated by adding the updated *pbest* particle if no particle from the archive dominates it.

After the steps mentioned above, a mutation operator with a given probability is applied, as

described by (Hamdan, 2009). The SMPSO algorithm is iterated for a certain number of times; the final result is then saved in the archive. Further details about the algorithm are available in (Nebro et al., 2009; Larabi-Marie-Sainte, 2011). The SMPSO pseudocode is shown in Algorithm 2.

Algorithm 2. SMPSO pseudocode
<pre> /*Initialization phase*/ For each particle { randomly generate position and velocity in the search space [0, 10]; particle.pbest is initialized to particle.position; /*Evaluation phase*/ For each objective function { evaluate particle; } } /*Saving phase (Archive)*/ Update the archive by adding the pbest; /*Iterations phase*/ iteration = 0; While iteration < maxIterations do { Select gbest; Calculate the velocity of each particle of the swarm (Equations 5- 10); Calculate the position of each particle of the swarm (Equations 3 & 4); Apply polynomial mutation; Evaluate each particle by calculating both objective functions (Equations 1 & 2); Update pbest; Update Archive; iteration ++; } Return the nondominated solutions (the archive) </pre>

4. Experiment Results and Discussion

In order to validate the effectiveness of the proposed SMPSO method, two different experiments have been conducted. In the first experiment, SMPSO was evaluated using UCI datasets and compared to state-of-the-art FS methods, namely NSGA-II, MOGA, and MOEA/D. In the second experiment, SMPSO was evaluated in the context of face recognition by using YALE face database.

4.1 Experiment 1: SMPSO for Feature Selection

The proposed FS technique and both objective functions were implemented in Java using a Dell XPS 9343 with an Intel® core TM i7-5500U CPU 2.40 GHz and 8-GB RAM. To compare the efficiency of the proposed multi-objective FS approach against state-of-the-art methods, four

UCI machine learning datasets were used, as described in Table 1 (Lichman, 2013).

Table 1. UCI datasets.

<i>Dataset Name</i>	<i># Features</i>	<i># Classes</i>	<i># Samples</i>
Iris	4	3	150
Glass	9	6	214
Wine	13	3	178
Vehicle	18	4	846

The choice of the parameters' values was delicate and depended on the dataset size. After several experiments on the proposed algorithm and considering the recommendations of (Durillo et al., 2009), the parameters were set, as it is shown in Table 2.

Table 2. Setup for SMPSO

<i>Parameter</i>	<i>Value</i>
Number of particles	100
Generation number	100
Number of runs	50
Archive size	50
The search space dimension	Features' number
The search space upper bound	10
The search space lower bound	1
C1 & C2	1.5+Random(0,1)
ω	Random(0,1)*0.4 + 0.1
Mutation probability	1/ Features' number

The generation number can be reduced if the dataset is small. Each experiment was executed 50 times to ensure that the obtained non-dominated solutions were the best ones and that they could not be further improved. The benefit of MOO approaches lies in providing many non-dominated solutions. These solutions focus on either optimizing Objective1 (intra-class) or on optimizing Objective2 (inter-class) or both. Figure 2 displays the Pareto Front for each dataset. The y-axis represents the Objective1, and x-axis represents the Objective2. As it is shown in the figure, the non-dominated solutions (the vectors of selected features) are uniformly distributed on the Pareto Front, which confirms the effectiveness of the abovementioned method (Durillo et al., 2009).

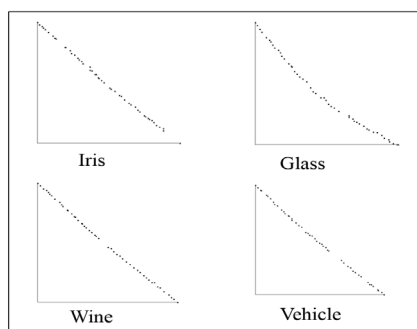


Figure 2. Pareto Front obtained after applying MOPSO to the four UCI machine learning datasets

In this study, three non-dominated solutions (Sol 1, Sol 2, Sol 3) were selected from the archive of each dataset, such that Sol 1 mainly optimized Objective1, Sol 2 mainly optimized Objective2, and Sol 3 was a compromise between optimizing both objective functions. For example, the three solutions obtained using the Iris dataset are illustrated in Figure 3. These solutions were later used to determine the best classification accuracy of each dataset.

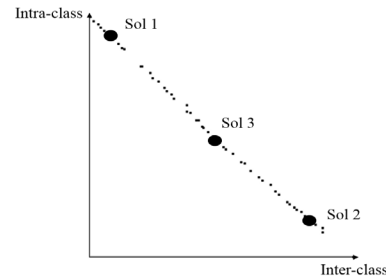


Figure 3. Three selected solutions represented on the Pareto Front of the Iris dataset

Table 3 displays, for each of the selected solutions obtained from the Iris database, the objective value of both objective functions, and the selected features. The selected features differ depending on the solution one focuses on. However, Sol 2 provides the maximum number of selected features, while Sol 1 always gives the minimum number of features since the inter-class variations need more features than intra-class variations. The above-mentioned process was applied to all the available datasets, however, in order to save space, only some of the results obtained were displayed.

Table 3. Results for three selected non-dominated solutions obtained by MOPSO using Iris dataset

<i>Solutions</i>	<i>Sol 1</i>	<i>Sol 2</i>	<i>Sol 3</i>
Obj 1	-483102.61	-5056.5612	-224724.68
Obj 2	720335.57	8068.6509	314266.34
Selected features	1, 4	All (1,2,3,4)	3, 4

To validate the results of the proposed MOPSO Feature Selection, a k-NN classifier is applied to the three selected solutions for each dataset, and only the solution with the best classification accuracy is addressed for further analysis and comparisons. In this experiment, k-NN was executed with tenfold cross-validation on the training set (80% of the whole dataset) after reducing the number of features with SMPSO. The value of the parameter k was selected between 5 and 50 based on the highest accuracy obtained in the training stage. When the model was trained with the optimal parameters, the test set (20% of the dataset) was employed for determining the classification

performance. For each of the UCI datasets, the k-NN parameter, the classification accuracy, and precision are indicated in Table 4. The proposed approach yielded promising classification accuracy rates compared to the baseline results which were obtained for the original feature space without dimensionality reduction.

Table 4. Validation of SMPSO using k-NN

Dataset		Iris	Glass	Wine	Vehicle
Baseline	<i>k value</i>	11	5	10	5
	<i>Accuracy (%)</i>	93.33	76.74	97.22	69.82
	<i>Precision (%)</i>	93.33	60	97.62	69.73
SMPSO	<i>k value</i>	8	5	5	5
	<i>Accuracy (%)</i>	96.67	83.72	100	70.41
	<i>Precision (%)</i>	96.67	62.7	100	70.34

Several comparisons were conducted against three well-known multi-objective FS methods, namely NSGA-II (Tan, Lim et Cheah, 2014), MOGA (Venkatadri & Srinivasa, 2010), and MOEA/D (Paul & Das, 2015), using the aforementioned UCI datasets. Table 5 shows

the results of SMPSO, NSGA-II, and MOEA/D. The classification accuracy is obtained using a 5-NN classifier for NSGA-II and MOEA/D. As it is shown, the classification accuracy was enhanced by using SMPSO compared to both FS methods. Additionally, the execution time of SMPSO was insignificant. However, the execution time of the other methods was not mentioned because of the differences among the execution environments.

Table 6 displays the results of SMPSO and MOGA. In a MOGA FS study (Venkatadri & Srinivasa, 2010), ten combinations of five measures of the objective functions were used, including attribute class correlation (AC), inter-class distance (IE), inconsistent example pairs (IP), Laplacian score (LS), and representation entropy (RE). In the aforementioned study, the classification accuracy was obtained using the Decision Tree classifier and Weka (J48). To provide a fair comparison, the same classifier/software was used in this experiment. A rank-based method was first applied in order to sort the classification precision obtained by MOGA and SMPSO from 1 to 11 for each dataset, where 1 represented the highest classification accuracy. Second, an average was calculated

Table 5. Comparison of SMPSO with NSGA-II and MOEA/D

	NSGA-II		MOEA/D		SMPSO		
	# Features	Accuracy (%)	# Features	Accuracy (%)	# Features	Accuracy (%)	Time (s)
Iris	2.00	96.03	2.00	97.27	2.00	96.67	1.26
Glass	7.40	66.77	4.40	67.76	6.32	83.72	1.24
Wine	11.2	95.90	6.90	96.05	8.8	100	1.26
Vehicle	17.3	68.06	9.10	65.26	11.86	70.41	1.40

Table 6. Comparison of SMPSO and MOGA (Venkatadri & Srinivasa, 2010).

FS method	Vehicle		Wine		Average of Accuracy Rank
	Accuracy	Rank	Accuracy	Rank	
IE+AC	72.43	5	93.24	2	3.5
IE+IP	74.00	2	92.12	3	2.5
IE+RE	73.96	3	92.12	4	3.5
IE+LS	72.74	4	89.35	7	5.5
AC+IP	69.99	6	89.93	6	6
AC+RE	63.84	9	90.02	5	7
AC+LS	52.49	10	83.10	10	10
IP+RE	67.62	8	77.48	11	9.5
IP+LS	68.44	7	89.31	8	7.5
RE+LS	52.49	11	87.61	9	10
MPSO	75.29	1	95.50	1	1

for the ranks allocated to different datasets. The method that generated the best results was the one with the lowest average rank. As it is shown in Table 6, the proposed approach proved to be the best FS method, with an average equal to 1. Again, SMP SO shows its efficient use as an FS approach for solving the curse of dimensionality problem.

4.2 Experiment 2: SMP SO for Face Recognition

To confirm the usefulness of the proposed approach for face recognition, experimentation was conducted using the YALE face database provided by the YALE Center for Computational Vision and Control (Belhumeur, Hespanha & Kriegman, 1997). The database consists of 15 users (i.e., classes), each of them with 11 face images with different variations in expression and illumination. Five user face images were used for training, and the remaining ones were kept for testing. These images were clipped to remove needless information and preserve only the facial components (e.g., eyebrows, eyes, nose, and lips). The size of the cropped image is 200x160 pixels.

In this experiment, the Laplacian smoothing transform (LST) was first applied to extract the facial features, as described by (Moujahdi et al., 2014). The extracted feature vector with a length of 40, which was named LST-40, was employed to select the pertinent features by using the SMP SO technique. The goal of this experiment was to show the classification performance when the number of features was reduced, and to demonstrate the efficiency of the proposed technique in selecting the salient and relevant features. To this end, a set of non-dominated solutions with an equal or different number of features was evaluated using a SVM classifier. For SVM classification, the library, *Libsvm*, was employed, which was implemented in *Matlab* by (Chang & Lin, 2011). The tenfold cross-validation was employed for selecting the optimal SVM parameters where (kernel type: Radial Basis Function, kernel parameters: $\gamma=0.025$, constant $C=100$).

Table 7 shows, for each non-dominated solution, the number of features and the classification rate when SMP SO is applied to the above-mentioned extracted feature set, that is LST-40.

Table 7. Classification Accuracy of the selected features after application of SMP SO on LST-40.

# Features	Accuracy (%)
8	83.33
10	86.67
11	90.30
13	86.36
14	89.40
15	86.97
16	87.57
17	88.24
18	85.15
19	87.87
20	89.09
21	86.67
22	89.70

The experimental results in Table 7 show that the accuracy rate is insufficient when the number of features is small (i.e., 8), which means that the eight selected features are not salient enough, and further features are needed. However, the accuracy rate is acceptable when the number of features is between 10 and 22. The highest accuracy rate (equal to 90.30 %) is obtained when the number of selected features is equal to 11. So, the proposed method improved classification performance by 3.35 %, compared to the classification rate of the complete feature set LST-40 (equal to 87.27 %).

This result shows the importance of obtaining diverse solutions by optimizing more than one objective. Moreover, it shows that reducing the number of selected features does not always ensure a higher classification accuracy rate but that an optimal number of features is needed, however. Therefore, it can be concluded that SMP SO can determine the relevant feature set that maximizes the accuracy rate and minimizes the computation complexity.

Further comparisons have been carried out against state-of-the-art approaches, including PCA (Eigenface), LDA (Fisherface), LPP (Laplacianface), OLPP (O-Laplacianface), and baseline methods. Table 8 illustrates the comparative results related to the implementation of the proposed SMP SO and to the best performance rate and the optimal dimensionality obtained by the state-of-the-art approaches obtained for the aforementioned YALE database, as it is shown by (He et al., 2005).

Table 8. Comparative results obtained by implementing different face recognition approaches for the YALE database.

Method	# of features	Error Rate (%)
Baseline	1024	45.6
Eigenface	71	45.2
Fisherface	14	22.5
Laplacianface	14	21.7
O-Laplacianface	14	17.9
Our proposed	11	9.7

As it can be seen in the table above, the proposed approach outperformed all the other above-mentioned methods in terms of the number of features and the classification error rate. The obtained results prove that SMPSO is a promising approach for reducing the feature vectors' dimensionality by selecting the most pertinent and salient features for face recognition. It is worth mentioning that Eigenface shows the worst performance since it does not provide an optimal representation of discriminative information.

5. Conclusion

FS is required before classification and clustering processes in order to decrease the size of feature vectors. Its use increases the class/cluster identification performance. It divides pertinent features into well-separated classes by reducing the problem of dimensionality. Several FS methods have been developed for handling face recognition. These methods are based on distinct measures that can be illustrated as measures of consistency, dependency, distance, information, and precision. These methods deal with the problem mentioned above as a single objective for identifying relevant features. However, the classification performance is not always

outstanding, because FS has to optimize two critical conflicting objectives, namely enhancing the classification accuracy and reducing the number of features. In this paper, a new multi-objective PSO-based FS method was proposed in order to determine relevant features of users' face templates, so that the intra-class variation was minimized and the inter-class variation was maximized. To the best of one's knowledge, PSO has only been used once in solving FS problems for face recognition. A thorough analysis and testing were conducted to investigate the efficiency of the proposed approach using different benchmark databases. The results of this approach were successfully compared to those obtained by implementing three of the existing optimization methods. SMPSO provided better classification accuracy rates in minimum time. Additionally, SMPSO was tested for the YALE face recognition dataset and proved successful in improving the classification performance.

As a recommendation for possible future works, SMPSO-based FS should be improved by varying the mutation operator and/or adding a tournament operator to increase the diversity of the obtained solutions. Moreover, initial solutions could be generated via a heuristic method instead of being generated randomly. Furthermore, the result could be improved by adding another objective in order to minimize the number of features. Although the experiment time was negligible, it can be further reduced by adequately implementing the inter- and intra-class variation distances. Finally, deep learning is an interesting approach to face recognition as it does not require feature selection. However, this approach should only be applied for very large and high-dimensional datasets.

REFERENCES

- Andreica, M. I., Briciu, S. & Andreica, M. E. (2009). Algorithmic Solutions to Some Transportation Optimization Problems with Applications in the Metallurgical Industry, *Metalurgia International*, 14(5), 46-53.
- Annarapu, C. S. R., Dara, S. & Banka, H. (2016). Cancer microarray data feature selection using multi-objective binary particle swarm optimization algorithm, *EXCLI Journal*, 15, 460-473.
- Bedia, P., Bansala, R. & Sehgal, P. (2012). Multimodal Biometric Authentication using PSO based Watermarking, *Procedia Technology*, 4, 612-618.
- Belhumeur, P. N., Hespanha, J. P. & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), 711-720.

- Cai, D., He, X., Han, J. & Zhang, H.-J. (2006). Orthogonal Laplacianfaces for Face recognition, *IEEE Transactions on Image Processing*, 15(11), 3608-3614.
- Chang, C. C. & Lin, C. J. (2011). LIBSVM: a library for support vector machines, *ACM Transactions on Intelligent Systems and Technology*, 2(3), 1-27.
- Cheng, J., Liu, Q., Lu, H. & Chen, Y. W. (2005). Supervised kernel locality preserving projections for face recognition, *Neurocomputing*, 67, 443-449.
- Clerc, M. & Kennedy, J. (2002). The particle swarm - explosion, stability, and convergence in a multidimensional complex space, *IEEE Transactions on Evolutionary Computation*, 6(1), 58-73.
- De-la-Hoz, E., De-la-Hoz, E., Ortiz, A., Ortega, J. & Martinez-Alvarez, A. (2014). Feature Selection by Multi-objective Optimization: Application to network anomaly detection by hierarchical self-organizing maps, *Knowledge-Based Systems*, 71, 322-338.
- Dinu, S. (2015). Multi-objective Assembly Line Balancing Using Fuzzy Inertia-adaptive Particle Swarm Algorithm, *Studies in Informatics and Control*, 24(3), 283-292. DOI: 10.24846/v24i3y201504
- Durillo, J. J., Garc a-Nieto, J., Nebro, A. J., Coello, C. A. C., Luna, F. & Alba, E. (2009). Multi-objective particle swarm optimizers: An experimental comparison. In *5th International Conference on Evolutionary Multi-Criterion Optimization* (pp. 495-509). Springer, Berlin, Heidelberg.
- Ekbal, A., Saha, S. & Garbe, C. S. (2010). Feature Selection Using Multiobjective Optimization for Named Entity Recognition. In *International Conference on Pattern Recognition* (pp. 1937-1940). IEEE.
- Ekbal, A. & Saha, S. (2012). Multiobjective optimization for classifier ensemble and feature selection: an application to named entity recognition, *International Journal on Document Analysis and Recognition*, 15(2), 143-166.
- Hamdan, M. (2010). On the disruption-level of polynomial mutation for evolutionary multiobjective optimization algorithms, *Computing and Informatics*, 29(5), 783-800.
- Hamdani, T. M., Won, J. M., Alimi, A. M. & Karray, F. (2007). Multi-objective Feature Selection with NSGA II. In *International Conference on Adaptive and Natural Computing Algorithms* (pp. 240-247). Springer, Berlin, Heidelberg.
- He, X., Yan, S., Hu, Y., Niyogi, P. & Zhang, H. J. (2005). Face recognition using Laplacianfaces, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 328-340.
- Jafri, R. & Arabnia, H. R. (2009). A survey of face recognition techniques, *Journal of Information Processing Systems*, 5(2), 41-68.
- Jain, A. K., Duin, R. P. W. & Mao, J. (2000). Statistical pattern recognition: A review, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1), 4-37.
- Larabi-Marie-Sainte, S. (2011). *Biologically inspired Algorithms for Exploratory Projection Pursuit*. PhD Thesis. Toulouse 1 Capitole University, June.
- Larabi-Marie-Sainte, S. (2015). Spatial Projection Pursuit based on Multiobjective Optimization. In *International Conference on Information and Communication Systems* (pp. 154-159). IEEE.
- Lichman, M. (2013). *UCI Machine Learning Repository*. Available at: <<http://archive.ics.uci.edu/ml>>. Irvine, CA: University of California, School of Information and Computer Science.
- Mandal, J. K., Mukhopadhyay, S. & Dutta, P. (editors) (2018). *Multi-Objective Optimization: Evolutionary to Hybrid Framework*. Springer.
- Moujahdi, C., Ghouzali, S., Mikram, M., Abdul, W. & Rziza, M. (2014). Inter-communication classification for multi-view face recognition, *International Arab Journal of Information Technology*, 11(4), 387-395.
- Muthukumar, A., Kasthuri, C. & Kannan, S. (2012). Multimodal Biometric Authentication Using Particle Swarm Optimization Algorithm with Fingerprint and Iris, *International Journal of Image and Video Processing*, 2(3), 369-374.
- Nebro, A., Durillo, J. J., Garc a-Nieto, J., Coello, C. A. C., Luna, F. & Alba, E. (2009). SMPSO: A New PSO-based Metaheuristic for Multi-objective Optimization. In *IEEE Symposium on Computational Intelligence in Multicriteria Decision Making* (pp. 66-73). IEEE.
- Paul, S. & Das, S. (2015). Simultaneous feature selection and weighting – An evolutionary multi-objective optimization approach, *Pattern Recognition Letters*, 65, 51-59.

- Rouhani, S., Fathian, M., Jafari, M. & Khavan, P. (2010). Solving the Problem of Flow Shop Scheduling by Neural Network Approach, In *International Conference on Networked Digital Technologies, Communications in Computer and Information Science*, 88 (pp. 172-183). Springer, Berlin, Heidelberg.
- Sheikholeslami, F. & Navimipour, N. J. (2017). Service allocation in the cloud environments using multi-objective particle swarm optimization algorithm based on crowding distance, *Swarm and Evolutionary Computation*, 35, 53-64.
- Soyel, H., Tekguc, U. & Demirel, H. (2011). Application of NSGA-II to feature selection for facial expression recognition, *Computers and Electrical Engineering*, 37(6), 1232-1240.
- Tan, C. J., Lim, C. P. & Cheah, Y. N. (2014). A multi-objective evolutionary algorithm-based ensemble optimizer for feature selection and classification with neural network models, *Neurocomputing*, 125, 217-228.
- Venkatadri, M. & Srinivasa, R. K. (2010). A Multiobjective Genetic Algorithm for Feature Selection in Data Mining, *International Journal of Computer Science and Information Technologies*, 1(5), 443-448.
- Xue, B., Zhang, M. & Browne, W. N. (2013). Particle Swarm Optimization for Feature Selection in Classification: A Multi-Objective Approach, *IEEE Transactions on Cybernetics*, 43(6), 656-671.
- Yang, J., Frangi, A. F., Yang, J., Zhang, D. & Jin, Z. (2005). KPCA Plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2), 230-244.
- Yann, C. & Siarry, P. (2013). *Multiobjective optimization: principles and case studies*. Springer.
- Zhang, Y., Gong, D. & Cheng, J. (2017). Multi-objective Particle Swarm Optimization Approach for Cost-based Feature Selection in Classification, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 14(1), 64-75.