

---

## Classification of Facial Emotions using Guided Particle Swarm Optimization I

---

Bashir Mohammed Ghandi\*

School of Mechatronics Engineering  
Universiti Malaysia Perlis (UniMAP)  
02600 Jejawi, Perlis, Malaysia  
E-mail: bmghandi@gmail.com  
\*Corresponding author

Ramachandran Nagarajan

School of Mechatronics Engineering  
Universiti Malaysia Perlis (UniMAP)  
02600 Jejawi, Perlis, Malaysia  
E-mail: nagarajan@unimap.edu.my

Hazry Desa

School of Mechatronics Engineering  
Universiti Malaysia Perlis (UniMAP)  
02600 Jejawi, Perlis, Malaysia  
E-mail: hazry@unimap.edu.my

**Abstract:** This paper presents a novel approach to facial emotion detection using a modified Particle Swarm Optimization algorithm, which we called Guided Particle Swarm Optimization (GPSO). The approach involves tracking the movements of 10 Action Units (AUs) placed at appropriate points on the face of a subject and captured in video clips. Two dimensional rectangular domains are defined around each of the AUs and Particles are then defined to have a component in each domain, effectively creating a 10-dimensional search space within which particles fly in search of a solution. Since there are more than one possible target emotions at any point in time, multiple swarms are used, with each swarm having a specific emotion as its target. At each frame in the video clip, the solution of the swarm that is nearest to its target is accepted as the solution. Our results so far show the approach to have a promising success rate.

**Keywords:** emotion detection; particle swarm optimization; PSO; facial emotions; facial expressions; facial action units.

**Reference** to this paper should be as follows: Ghandi, B. M. Nagarajan, R., Desa, H., "Classification of Facial Emotions using Guided Particle Swarm Optimization I", *Int. J.CCT*, Vol.1, No. 1, pp. 36-46.

**Biographical notes:** Bashir Mohammed Ghandi received his B.Sc in Mathematics in 1988 at Usman Danfodiyo University, Sokoto, Nigeria. He obtained his MSc in Mathematics at University of Ibadan, Nigeria in 1991. He obtained MSc in Computer Science at University of Salford, UK. in 1994. He has over 15 years experience as a university lecturer in Nigeria and in Saudi Arabia. He is currently a PhD student in the School of Mechatronics Engineering, Universiti Malaysia Perlis. His research interests include computational intelligence, machine vision, computer networks and object-oriented programming. He is a Student Member of IEEE.

Ramachandran Nagarajan obtained his BE (Hons), M.Tech and Ph D degrees from Madras University, IIT Kanpur and Madras University, India, respectively. He has worked in teaching and research positions in a

number of universities in India, Iraq, Malaysia, the UK and Japan. He is currently with University Malaysia Perlis (UniMAP), Malaysia, as a Professor in the School of Mechatronics Engineering. He has been working in the fields of Robotics and Intelligent Control for several years. He has several contributions as peer reviewed Journal and Conference papers, Books, Book chapters, monographs and documented research reports. He has offered invited lectures in universities of India, Malaysia, Japan and Taiwan. His current fields of interest are in Hospital Robots, Emotion Controlled Machines and Robot based SLAM. He is a Life Fellow of IE (India), Senior Member of IEEE (USA), Member, Association of Biomedical Soft computing (BMSA), Japan, and a Member, IET (UK).

Hazry Desa received the B.S.Eng in Mechanical Engineering from Tokushima University, Japan in 1997. He was a Mechanical Senior Design Engineer in R&D Sony Technology Malaysia. From 2001 to 2003 he was employed by the Tamura Electronics Malaysia. In October 2003 he joined the Northern Malaysia University College Of Engineering, Malaysia as a fellow and at the same time he worked towards his Ph.D. degree in the Electrical and Electronics Dept. Oita University, Japan. After completing his study, he was appointed as the Deputy Dean at the School of Mechatronic, University Malaysia Perlis and currently he is the Dean of Centre for Communication Skills and Entrepreneurship. He is also the Head of the Research Cluster for Autonomous System and Machine Vision (Auto-Mav). His research interests include robotics, nonholonomic systems and image processing. He is a member of IEEE.

---

## 1 Introduction

Human beings interact with one another not only using verbal languages but also using nonverbal methods such as gestures and facial expressions or emotions. Recently, a lot of research effort has been extended towards improving human-computer interaction so that computers can also have the intelligence to perceive the emotional state of a human and react accordingly. There are several applications that can be derived from such technology. For example, intelligent mechanisms, such as robots could be developed to assist bed-ridden and highly disabled people who are confined to a room in their houses. This is important given the present modern life style where the population of children is declining, the middle-aged are getting busier with work schedules and where the senior citizens and the disabled are increasingly being left to fend for themselves. Thus, in such circumstances, the development of welfare robots that are able to perceive emotions and act accordingly could be a means of providing support and comfort to the senior citizens and the disabled for the rest of their lives.

A recent algorithm that has been found to be very efficient and effective in solving a variety of problems that involves optimization or searching is the Particle Swarm Optimization (PSO) algorithm. PSO is a population-based search algorithm that was first developed by Eberhart and Kennedy [1], whose initial intent was to simulate the social behaviour of birds as they fly in a group searching for food. PSO either in its original form or with some modifications was soon found to be applicable in solving a variety of problems. Examples of its applications include the classical travelling salesman problem [2], electrical power systems [3], neural networks training [4][5] and solving systems of equations [6]. It has been applied to clustering problems such as image clustering [7], data clustering [8], data mining [9] and gene clustering [10]. Other applications of PSO are in the areas of underwater acoustics [11], internal combustion engines [12], antenna designs [13], human tremor analysis [14], task assignment [15] and combinational logic circuits design [16], etc. However, to our knowledge, PSO has not been applied directly in solving emotion detection problems. In this paper we present a modified version of the algorithm that we called, Guided Particle Swarm Optimization (GPSO) and which we successfully applied in detecting facial emotions with promising success rates.

The rest of this paper is organized as follows: Section 2 discusses emotion detection, where we identified some of the methods researchers have used in tackling the problem. We concluded the section by outlining our own approach to the emotion detection problem. In

section 3 we introduced the original PSO algorithm and then explained the GPSO, which is our own modification to the algorithm that is designed to solve emotion detection problem. In section 4 we presented and discussed our preliminary results. Finally, in section 5, we present our conclusions and identify future research works that we intend to carry out.

## 2 Detection of facial emotions

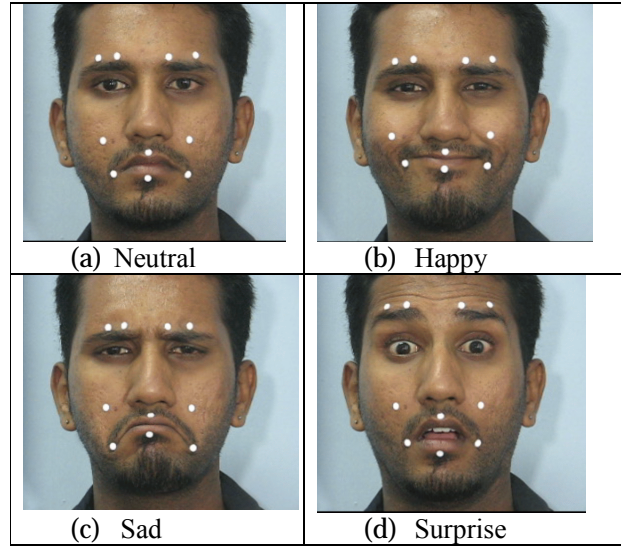
Discrete emotion theorists have identified six basic emotions that are universally expressed and recognized independently from cultural background. These include happiness, anger, sadness, surprise, disgust and fear [17]. There are many more types of emotions that are expressed by people such as ‘boredom’, ‘I don’t know’, etc. However, there is much less evidence that these subtler expressions are universally displayed and interpreted [18].

One approach to facial expressions classification is to recognize the underlying facial muscle activities and then interpret these in terms of arbitrary categories such as emotions, attitudes or moods [19]. The Facial Action Coding System (FACS) [20] is the best known and the most commonly used system developed for human observers to describe facial activity in terms of visually observable facial muscle actions (i.e., Action Units, AUs). With FACS, human observers uniquely decompose a facial expression into one or more of 44 AUs that produced the expression in question [18]. Recent work on emotion detection based on biologically inspired algorithms has used ANNs [21], SVMs [22], Bayesian Networks [23][24] and Hidden Markov Models (HMMs) [23]. Recent work on facial AU detection applying biologically inspired algorithms has used similar techniques: ANNs [25], SVMs [22][18] and Bayesian Networks [24]. For a survey of past work in the field, see [19].

Our own methodology is based on studying the underlying AUs that are involved in expressing the different types of emotions. We identify the specific AUs whose movements we wish to observe using small luminous markers that are placed on the face of the subject. A video clip of the subjects is then recorded as they expressed different types of emotions. Figure 1 shows some sample shots from the video clip recorded on one of our subjects. Our aim is to identify the emotion being expressed at each frame in the video clip by simply observing the changes in the positions of the AUs.

Once we have a video clip of a subject, the first step in our emotion detection process is to digitize the clip to obtain the positions of the AUs in terms of x, y coordinates over time. Figure 2 shows a small portion of a sampled data file resulting from digitizing a video clip. The second step in our experiment is to go through a training session for a particular subject. In this session, we manually teach our program (see details in section 5) the approximate positions of the AUs for each of the emotions we wish to detect. Finally, the program is executed for the full length of the video clip where it visually displays the emotion being expressed at each frame of the clip on a continuous basis. The program itself is a direct implementation of GPSO, which is our modification to the basic PSO algorithm that is designed for the purpose of emotion detection. We discuss PSO and GPSO in the next section.

**Figure 1** Positions of AUs in different emotions



**Figure 2** A small cut-out from a sample video data file obtained after digitization.

Frame #	Time	Point #1 X	Point #1 Y	Point #2 X	Point #2 Y	Point #3 X	Point #3 Y	Point #4 X	Point #4 Y	Point #5 X	Point #5 Y
1	0.143	255.718	377.344	320.235	382.031	451.613	377.344	518.475	385.547	283.871	173.
2	0.286	256.393	377.000	320.961	381.500	452.205	378.000	519.221	386.000	284.699	173.
3	0.429	256.305	376.500	320.973	381.000	452.200	378.000	519.358	385.000	284.807	172.
4	0.571	256.406	376.500	320.981	381.000	452.493	378.000	519.410	385.000	284.722	173.
5	0.714	256.862	377.000	321.609	381.500	453.109	378.000	520.133	385.000	285.460	173.
6	0.857	257.526	377.000	322.210	381.500	453.807	378.000	520.648	385.000	285.700	173.
7	1.000	257.402	377.000	322.046	381.000	453.505	378.000	520.482	385.000	285.547	173.
8	1.143	256.938	376.500	321.587	381.000	453.003	378.000	520.116	385.000	285.311	172.
9	1.286	256.815	376.500	321.436	381.000	452.959	378.000	520.090	385.000	285.256	172.
10	1.429	256.611	376.500	321.415	381.000	452.926	378.000	519.958	385.000	285.000	172.

### 3 PSO and GPSO

#### 3.1 Particle Swarm Optimization (PSO)

PSO is a population-based search algorithm designed initially to simulate the social behaviour of birds in a flock as they fly in search of food. A PSO algorithm maintains a swarm of particles, where each particle represents a potential solution [26]. Particles are “flown” through a multi-dimensional search space, where the position of a particle is adjusted according to two factors:

- its own successful experience
- The successful experiences of its neighbours.

Let  $x_i(t)$  denote the position of particle  $i$  at time  $t$ . The position of the particle is changed by adding a velocity,  $v_i(t+1)$  to the current position.

$$x_i(t + 1) = x_i(t) + v_i(t + 1) \tag{1}$$

where  $x_i(0)$  is generated randomly from the range  $[x_{min}, x_{max}]$

It is the velocity vector that drives the optimization process, and reflects both the experience of the particle and the experiences of its neighbours. The experiential knowledge of the particle is referred to as the *cognitive* component, and is proportional to the distance of the particle from its own best position [26].

The socially exchanged information is referred to as the *social* component of the velocity equation. Originally, two PSO algorithms have been developed, which differ in the size of their neighbourhoods. These two algorithms are known as *gbest* and *lbest* [26].

### 1.1.1 Global best PSO (*gbest*)

For the global best PSO, the neighbourhood for each particle is the entire swarm. The social networking employed by *gbest* PSO reflects the star topology, where the social component of the velocity equation reflects the information obtained from the entire swarm [26]. In this case, the social component is the best position found by the swarm, represented as  $\hat{y}(t)$ . For *gbest* PSO, the velocity of particle  $i$  is calculated as:

$$v_i(t+1) = v_i(t) + c_1 r_1(t)[y_i(t) - x_i(t)] + c_2 r_2(t)[\hat{y}(t) - x_i(t)] \quad (2)$$

where,  $v_i(t)$  is velocity of particle  $i$  in a given dimension at time  $t$ ,  
 $x_i(t)$  is the position of particle  $i$  in a given dimension at time  $t$ ,  
 $c_1$  and  $c_2$  are positive acceleration constants,  
 $r_1(t)$ ,  $r_2(t)$  are random values in the range  $[0, 1]$ , generated at time  $t$ ,  
 $y_i(t)$  is the best position so far found by particle  $i$ .

For minimization problem, the personal best at the next time step,  $t+1$ , is calculated as:

$$y_i(t+1) = \begin{cases} y_i(t) & \text{if } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{if } f(x_i(t+1)) < f(y_i(t)) \end{cases} \quad (3)$$

where  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  is the fitness (or objective) function, which measures how close the corresponding solution is to the optimum. The *gbest* PSO algorithm is summarized in Figure 3.

**Figure 3** PSO (Global best) algorithms.

1. Create and initialize an  $n$  – dimensional swarm;
2. repeat
3.     for each particle  $i = 1, \dots, n$  do
4.         //set the personal best position
5.         if  $f(x_i) < f(y_i)$  then
6.              $y_i = x_i$ ;
7.         end
8.         // set the global best position
9.         if  $f(y_i) < f(\hat{y})$  then
10.              $\hat{y} = y_i$ ;
11.         end
12.     end
13.     for each particle  $i = 1, \dots, n$  do
14.         update the velocity using equation (2);
15.         update the position using equation (1);
16.     end
17. until stopping condition is true

### 1.1.2 Local best PSO (*lbest*)

The local best PSO, *lbest*, uses a ring social network topology, where smaller neighbourhoods are defined for each particle [26]. The social component reflects the information exchanged within the neighbourhood of the particle. Thus, the velocity update equation is modified as follows:

$$v_i(t+1) = v_i(t) + c_1 r_1(t)[y_i(t) - x_i(t)] + c_2 r_2(t)[\hat{y}_i(t) - x_i(t)] \quad (4)$$

where  $\hat{y}_i(t)$  is the best position found by the neighborhood of particle  $i$  in a given dimension. The two versions of PSO algorithms are similar in the sense that the social component of the velocity updates causes both to move towards the global best.

There are two main differences:

- Due to the larger particle interconnectivity of *gbest*, it converges faster than *lbest*. This convergence comes at the cost of less diversity.
- Due to the larger diversity in *lbest*, which results in more coverage of the search space, it is less prone to being trapped in local minima.

In general, neighbourhood structures such as the ring topology used in *lbest* improves its performance [27]

## 3.2 Guided Particle Swarm Optimization (GPSO)

The emotion detection problem is essentially a search problem, where at each point; we are searching to identify which of the possible emotions does the current facial expression represents. Thus, clearly emotion detection lends itself as a possible candidate for PSO application, since PSO is basically a search algorithm. However, in order to apply PSO to solve the emotion detection problem, we need to first define the various parameters of the algorithm in relation to the problem. In particular, we need to define the following:

- a) What is the search space and what is its dimension?
- b) How do we represent a particle in the emotion-detection setting?
- c) How do we represent the position and velocity of a particle?
- d) What is the objective function being minimized by the PSO.

Recall that in section 2, we have stated our approach to the emotion detection problem, which is basically to monitor the changes in the positions of the action units, placed on the face of a subject over a period of time, from which we can then determine the emotion expressed at each point in time. With this in mind, we define the parameters of the PSO as follows:

*Definition 1:* Search space and its dimension

Let the Action Units (AUs), be denoted by,  $q_1, q_2, \dots, q_n$ .

Let  $D_1, D_2, \dots, D_n$  represent the domains of the AUs,  $q_1, q_2, \dots, q_n$  respectively. That is  $D_j$  represents a 2-dimensional rectangular window consisting of the possible points that  $q_j$  can be assigned to. Then the search space is a  $n$ -tuple,  $R^n$ , given by:

$$R^n = (D_1, D_2, \dots, D_n) \quad (5)$$

The dimension of the search space is  $n$ , where  $n$  is the number of action units being observed.

*Definition 2:* Particle, Position and Velocity:

A particle  $P$  is an abstract object in the  $R^n$  search space that has a position and a velocity and represents a possible solution.

The position,  $x_i(t)$  of a particle,  $P_i$  at time  $t$ , is a complete assignment of values ( $val_1, val_2, \dots, val_n$ ), where  $val_j \in D_j$ . Thus,  $x_i(t)$  is a vector, ( $val_1, val_2, \dots, val_n$ ).

The velocity,  $v_i(t)$  of particle  $i$  at time  $t$  is an  $n$ -tuple ( $v_1, v_2, \dots, v_n$ ) where  $v_j$  represents the velocity of the particle in dimension  $D_j$ .

There are two peculiar issues which make the emotion detection problem a little different than normal problems to which PSO is applied. First, in normal PSO problems, there is usually one target that all particles in the swarm are trying to reach. In our particular case however, there are a number of possible emotions and any one of them could be encountered at any time. In order to solve this multi-target problem, we propose to have multiple swarms, one for each possible emotion. Since each swarm has a different target to reach, the objective function of each swarm must be defined differently. We define the objective function of each swarm as the Euclidean distance between its current best position and its target.

For example, the following is the definition of the objective function for the swarm that is targeting the happy emotion.

Let  $S = (s_1, s_2, \dots, s_n)$  represent the happy emotion. (Note: this is derived through the training session). Then the objective function for the swarm,  $f_s : \mathbb{R}^n \rightarrow \mathbb{R}$ , is defined as:

$$f_s(X_i(t)) = |X_i(t) - S| \\ = \sqrt{(x_1 - s_1)^2 + (x_2 - s_2)^2 + \dots + (x_n - s_n)^2} \quad (6)$$

The objective functions for the other swarms are defined similarly.

Our proposal is that, in each iteration of the PSO algorithm, each swarm will update the positions of its particles as usual and obtains its best position. These best positions are then compared. The swarm whose best position is closest to its target is considered to have found a solution. For example if that swarm happens to be the happy-targeting swarm, then the current state of the video clip is identified as happy. We note that these computations are repeated for each frame in the video clip. Thus, the number of iterations for the algorithm is simply the number of frames in the video clip.

The second issue that makes the emotion-detection problem a little different than normal PSO problems is that in this case we have the data about the positions of the action units. If the particles can take advantage of this knowledge, then they are likely to reach their target sooner than if they rely solely on their *cognitive* and *experiential* knowledge. Accordingly, we propose the following changes to the algorithm:

- The positions of the AUs should always be represented as one of the particles in each swarm. That is, let  $Q$  be a particle whose position  $X_q(t) = (q_1, q_2, \dots, q_n)$ , where  $q_1, q_2, \dots, q_n$  are the positions of the  $n$  AUs respectively. Then  $Q$  must be included as a particle in each swarm.
- We change the velocity update equation from equation (2) so that the position of  $Q$  is always regarded as the global best. That is,

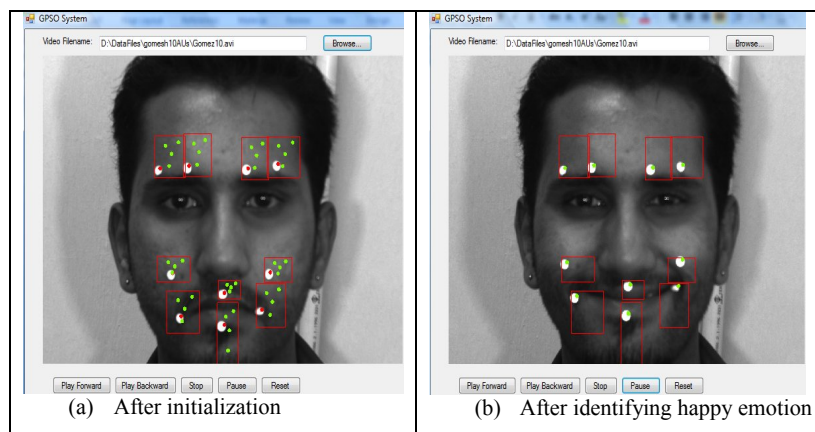
$$v_i(t+1) = v_i(t) + c_1 r_1(t)[y_i(t) - x_i(t)] + c_2 r_2(t)[q(t) - x_i(t)] \quad (7)$$

With these changes, it means the particles are guided to converge towards the path of the action units. Accordingly, we call this modified version of the algorithm the Guided Particle Swarm Optimization (GPSO) algorithm.

## 4 Experimental Results

The GPSO discussed in section 4 was implemented using C# programming language under the .NET development framework. The implemented program has two modes, the learning mode and the detection mode. In the learning mode, the user will run a video clip to capture the positions of the AUs corresponding to each of the basic emotion under study. Once a particular emotion is observed, the user will pause the video and then click the relevant button to save the current positions of the AUs into a file as the coordinate values for the emotion. The learning session is ended as soon as the data for all the relevant emotions is obtained. In the detection mode, the system will take as input a video clip, the digitized data for the video clip and the positions of the AUs corresponding to the various emotions as captured in the training session. The system initializes a swarm by creating random particles within the domain of each of the AUs (Figure 4(a)). The forward button is then clicked to run the video clip. This will trigger the execution of the GPSO algorithm, which will return for each frame, the detected emotion. The detected emotion is visually displayed on the screen. Due to the introduced modification to the algorithm, where particles are guided to converge towards the path of the AUs, it was observed that particles converge very quickly towards the AUs. Figure 4 shows the particles at the initial stage (Figure 4 (a)) and after particles converge to find the happy emotion (Figure 4(b)).

**Figure 4** Sample images from the GPSO system



For this preliminary study, we considered only three of the six universal basic emotions, namely happy, surprise and sad. These three, plus the neutral position, gives four possible states that the GPSO system can detect presently. We have tested the system with 6 different video clips of different subjects. Each video clip was about 30 seconds long or of 200 frames, displaying at about 7 frames per second. In order to test the performance of the detection algorithm, we have made the system to pause at each frame, where we manually identify the emotion being displayed. For each frame, the emotion that is automatically detected by the system and the one that is manually detected by the human user are recorded in a file. Table one shows the average success rates recorded after taking the data 10 times for each video clip. Success here means where the auto-detection and the manual detection coincide.



**Table 1** Results of the preliminary emotion detection by GPSO

<i>Video Clip</i>	<i>Number of Frames</i>	<i>Success</i>	<i>% Success</i>
Video Clip # 1	200	170	85%
Video Clip # 2	200	184	92%
Video Clip # 3	200	172	86%
Video Clip # 4	200	180	90%
Video Clip # 5	200	175	87.5%
Video Clip # 6	200	190	95%

As shown in Table 1, the success rates recorded ranges from 85% to 95%. Clearly these are promising set of results. In fact these results were even better than they appeared to be because on close examination of the data files containing the results of the auto-detection and the manual detection, the errors were mainly found during transitions from neutral state to some emotion or from some emotion to the neutral state. In these transition states, even to the human user, it is really difficult to say exactly what the state is. Perhaps we should have another, 'unknown' state to categorize such cases. If that is done, then the results would certainly be much better.

## 5 Conclusion

In conclusion, we have presented a novel approach to emotion detection, where we showed how this can be achieved using a modified version of the Particle Swarm Optimization algorithm. By its very nature, PSO is a concurrent search algorithm where multiple particles are involved in searching different portions of the search space in parallel, thus increasing the chances of finding a solution sooner. Therefore implementing an emotion-detection algorithm that employs PSO is bound to be advantageous. Our experiments so far showed promising results in terms of accuracy of detection.

In terms of future research work on our project, we intend to closely look at the GPSO to see in what ways we can improve its performance both in terms of run-time efficiency and accuracy of detection. We also intend to extend the system to cover the other three basic emotions, namely fear, anger and disgust. The ultimate goal is to develop a system that can run on real-time video stream and to embed it into a robot for some practical useful purposes.

## Acknowledgments

The authors are thankful to Ministry of Science, Technology and Innovation, Malaysia, for funding through eScience, code: 9005-00020.

## References

- [1] Kennedy, J. and Eberhart, R. (1995) 'Particle swarm optimization' *Proceedings of IEEE International Conference on Neural Networks, Perth, Australia* pp. 1942-1948.
- [2] Wang, K. P., Huang, L., Zhou, C. G. and Pang, W. (2003) 'Particle Swarm Optimization for Travelling Salesman Problem', in *Proceedings of International Conference on Machine Learning and Cybernetics*, pp. 1583-1585.
- [3] Miranda, V. and Fonseca, N. (2002) 'EPSO -- Evolutionary Particle Swarm Optimization, a New Algorithm with Applications in Power Systems' *Proceedings of the Asia Pacific IEEE/PES Transmission and Distribution Conference and Exhibition*, vol. 2, pp. 745-750.
- [4] Al-Kazemi, B. and Mohan, C.K. (2002) 'Training Feedforward Neural Networks using Multi-Phase Particle Swarm Optimization', in *Proceedings of the Ninth International Conference on Neural Information Processing*, vol. 5, pp. 2615-2619.
- [5] Mendes, R., Cortez, P., Rocha, M. and Neves, J. (2002) 'Particle Swarms for Feed forward Neural Network Training' *Proceedings of the International Joint Conference on Neural Networks*, pp. 1895-1899.
- [6] Brits, R., Engelbrecht, A.P. and van den Bergh, F. (2002) 'Solving Systems of Unconstrained Equations using Particle Swarm Optimization' *Proceedings of the IEEE Conference on Systems, Man, and Cybernetics*, vol. 3, pp. 102-107.
- [7] Omran, M.G., Engelbrecht, A.P. and Salman, A. (2005) 'Particle Swarm Optimization Method for Image Clustering' *International Journal on Pattern Recognition and Artificial Intelligence*, vol. 19, no. 3, pp. 297-322.
- [8] van der Merwe, D.W. and Engelbrecht, A.P. (2003) 'Data Clustering using Particle Swarm Optimization' *Proceedings of IEEE Congress on Evolutionary Computation*, vol. 1, pp. 215-220.
- [9] Sousa, T., Silva, A. and Neves, A. (2004) 'Particle Swarm Based Data Mining Algorithms for Classification Tasks', *Parallel Computing*, vol. 30, no. 5-6, pp. 767-783.
- [10] Xiao, X., Dow, E.R., Eberhart, R.C., Ben Miled, Z. and Oppelt, R.J. (2003) "Gene Clustering using Self-Organizing Maps and Particle Swarm Optimization", in *Online Proceedings of the Second IEEE International Workshop on High Performance Computational Biology*.
- [11] Thompson, B.B., Marks, R.J., El-Sharkawi, M.A., Fox, W.J. and Miyamoto, R.T. (2003) Inversion of Neural Network Underwater Acoustic Model for Estimation of Bottom Parameters using Modified Particle Swarm Optimizer', in *Proceedings of the International Joint Conference on Neural Networks*, pp. 1301-1306.
- [12] Ratnaweera, A., Watson, H. and Halgamuge, S.K. (2003) 'Optimisation of Valve Timing Events of Internal Combustion Engines' *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 4, pp. 2411-2418.
- [13] Rahmat-Samii, Y., Gies, D. and Robinson, J. (2003) 'Particle Swarm Optimization (PSO): A Novel Paradigm for Antenna Designs' *The Radio Science Bulletin*, vol. 304, pp.14-22.
- [14] Eberhart, R.C. and Hu, X. (1999) 'Human Tremor Analysis using Particle Swarm Optimization' *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 3, pp. 1927-1930, July.
- [15] Salman, A., Ahmad, I. and Al-Madani, S. (2002) 'Particle Swarm Optimization for Task Assignment Problem', *Microprocessors and Microsystems*, vol. 26, no. 8, pp. 363-371.
- [16] Coello Coello, C.A., Luna, E.H.N. and Aguirre, A.H.N. (2003) 'Use of Particle Swarm Optimization to Design Combinational Logic Circuits' *Lecture Notes in Computer Science*, vol. 2606, pp. 398-409. Springer-Verlag.
- [17] Keltner, D. and Ekman, P. (2000) 'Facial expression of emotion', *Handbook of emotions*, M. Lewis and J.M. Haviland-Jones, Eds. Guilford Press, New York, pp. 236-249.
- [18] Valstar, M.F., Patras, I. and Pantic, M. (2005) 'Facial Action Unit Detection using Probabilistic Actively Learned Support Vector Machines on Tracked Facial Point Data' *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 3, pp. 76-84.
- [19] Pantic, M. and Rothkrantz, L.J.M. (2003) 'Toward an Affect-Sensitive Multimodal Human-computer Interaction' *Proceedings of the IEEE*, vol. 91, no. 9, pp. 1370-1390.
- [20] Ekman, P., Friesen, W.V. and Hager, J.C. (2002) 'The Facial Action Coding System: A Technique for the Measurement of Facial Movement' *San Francisco: Consulting Psychologist*.

- [21] Fasel, B., Monay, F. and Gatica-Perez, D. (2004) 'Latent semantic analysis of facial action codes for automatic facial expression recognition' *Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pp. 181-188.
- [22] Bartlett, M.S., Littlewort, G., Lainscsek, C., Fasel, I. and Movellan, J. (2004) 'Machine learning methods for fully automatic recognition of facial expressions and actions', *IEEE International Conference on Systems, Man and Cybernetics*, vol. 1, pp. 592-597.
- [23] Cohen, I., Sebe, N., Garg, A., Chen, L.S. and Huang, T.S (2003) 'Facial expression recognition from video sequences - temporal and static modelling' *Journal of Computer Vision and Image Understanding*, vol. 91, pp. 160-187.
- [24] Zhang, Y. and Ji, Q. (2005) 'Active and Dynamic Information Fusion for Facial Expression Understanding from Image Sequences' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 5, pp. 699-714.
- [25] Tian, Y., Kanade, T. and Cohn, J.F. (2001) 'Recognizing action units for facial expression analysis' *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 97-115.
- [26] Engelbrecht, A. P. (2007) 'Computational Intelligence: An Introduction', Second Edition, New York, NY: Wiley.
- [27] Kennedy, J. and Mendes, R. (2002) 'Population Structure and Particle Performance' *Proceedings of the IEEE congress on Evolutionary Computation*, pp. 1671-1676.