Proceedings of the International Conference on Man-Machine Systems (ICoMMS)
11 – 13 October 2009, Batu Ferringhi, Penang, MALAYSIA

Facial Emotion Detection using Guided Particle Swarm Optimization (GPSO)

Bashir Mohammed Ghandi, R. Nagarajan, Hazry Desa
School of Mechatronic Engineering, Universiti Malaysia Perlis (UniMAP), 02600 Jejawi, Perlis, Malaysia
bmghandi@gmail.com, {nagarajan, hazry}@unimap.edu.my

Abstract— **In this paper, we present a novel approach to human facial emotion detection by applying a modified version of the Particle Swarm Optimization (PSO) algorithm, which we called Guided Particle Swarm Optimization (GPSO). Our approach is based on tracking the movements of facial action units (AUs) that are placed on the face of a subject and captured in video clips. We defined particles that form swarms as vectors consisting of points from each domain of the AUs considered. Particles are allowed to move around the effectively n-dimensional search space in search of the emotion being expressed in each frame of a video clip (where n is the number of action units being tracked). 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. We have implemented and tested the algorithm on video clips that contain all the six basic emotions, namely happy, sad, surprise, disgust, anger and fear. Our results show the algorithm to have a promising success rate.**

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

I. Introduction

Human-Computer interaction is a very important and active research area where researchers are putting a lot of efforts to device methods and algorithms that allows computers to perceive the emotional state of the human user and react accordingly. There are several applications that can be derived from such technology. For example, intelligent welfare robots could be developed to provide support and comfort to 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.

A recent algorithm that has been found to be very efficient and effective in solving a variety of problems that involve 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 in 1995, whose initial intent was to simulate the social behavior of birds as they fly in a group searching for food [1]. 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 application include the classical travelling salesman problem [2], electrical power systems [3] and neural networks training [4]. It has been applied to clustering problems such as image clustering [5], data clustering [6] and Gene clustering [7]. Other applications of PSO are in the areas of underwater acoustics [8], task assignment [9] and combinational logic circuits design [10], 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 refer to as, Guided Particle Swarm Optimization (GPSO), which we successfully applied in detecting facial emotions with promising success rates.

The rest of this paper is organized as follows: Section II discusses emotion detection, where we identified some of the methods researchers have used in tackling the problem. In section III we introduce the original PSO algorithm and then explained the GPSO, which is our own modification to the algorithm designed for emotion detection. In section IV we present and discuss our results. Finally, in section V, we present our conclusions and identify future research work that we intend to carry out.

Emotion Detection

Six basic emotions have been identified in the literature to be universal and independent of cultural background, both in terms of how they are expressed and how they are perceived. These include happiness, anger, sadness, surprise, disgust and fear [11]. 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 expressions are universally displayed and interpreted [12].

One approach to facial expressions classification is to recognize the underlying facial muscle activities and then interpret these in terms of categories such as emotions, attitudes or moods [13]. The Facial Action Coding System (FACS) [11] 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 [12]. Recent work on facial AU detection applying biologically inspired algorithms has used: ANNs [14], SVMs [15], [16], and Bayesian Networks [17]. A good survey of past work in the field was presented in [13].

Our 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. Fig. 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 and analyzing the changes in the positions of the AUs.

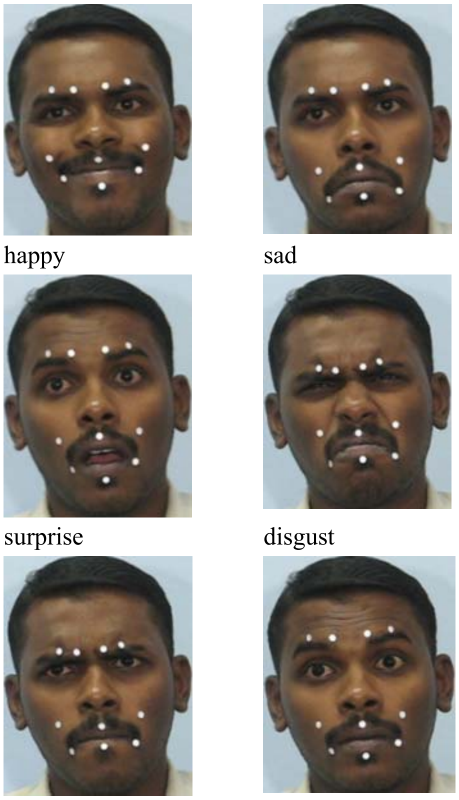

Fig. 1.Positions of AUs in different emotions

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. Fig. 2 shows a small portion of a sampled data file resulting from digitizing a video clip.

Frame #	Time	Point #1 X	Point #1 Y	Point #2 X	Point #2 Y	Point #3 X	Point #3 Y
1	0.143	255.718	377.344	320.235	382.031	451.613	377.34
2	0.286	256.393	377.000	320.961	381.500	452.205	378.00
3	0.429	256.305	376.500	320.973	381.000	452.200	378.00
4	0.571	256.406	376.500	320.981	381.000	452.493	378.00
5	0.714	256.862	377.000	321.609	381.500	453.109	378.00
6	0.857	257.526	377.000	322.210	381.500	453.807	378.00
7	1.000	257.402	377.000	322.046	381.000	453.505	378.00
8	1.143	256.938	376.500	321.587	381.000	453.003	378.00
9	1.286	256.815	376.500	321.436	381.000	452.959	378.00
10	1.429	256.611	376.500	321.415	381.000	452.926	378.00

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

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 IV) 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 we designed for the purpose of emotion detection. We discuss PSO and GPSO in the next section.

PSO and GPSO

Particle Swarm Optimization (PSO)

PSO is a population-based search algorithm initially designed to simulate the social behavior 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 [18]. 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 neighbors.

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.

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

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 neighbors. 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 [18]. The socially exchanged information is referred to as the *social* component of the velocity equation. Originally, two PSO algorithms were developed, which differ in the size of their neighborhoods. These two algorithms are known as *gbest* and *lbest* [18].

For the global best PSO, the neighborhood 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 [18]. 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 in (2).

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

where, $v_i(t)$ is the velocity of particle i in a given dimension at time t. $x_i(t)$ is the position of 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 and $y_i(t)$ 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:

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

where $f : \mathrm{R}^n \longrightarrow \mathrm{R}$ is the fitness (or objective) function, which measures how close the corresponding solution is to the optimun. Fig. 3. Summarizes the *gbest* PSO algorithm.

The local best PSO, *lbest*, is similar to the *gbest,* except that it uses a ring social network topology, where smaller neighborhoods are defined for each particle [18]. The social component reflects the information exchanged within the neighborhood of the particle. Thus, the velocity update equation is modified as in equation (4).

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

where $\hat{y}_i(t)$ is the best position found by the neigborhood of particle i in a given dimension.

```
Create and initialize an n – dimensional swarm;
repeat
for each particle i = 1, . . ., n do
  //set the personal best position
  if f(xi) < f(yi) then
      yi = xi;
  end
  // set the golobal best position
  if f(yi) < f(ŷ) then
      ŷ = yi;
  end
end
for each particle i = 1, ..., n do
  update the velocity using equation (2);
  update the position using equation (1);
end
until stopping condition is true
```

Fig. 3.PSO (Global best) algorithm.

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, neighborhood structures such as the ring topology used in *lbest* improves its performance [19].

Guided Particle Swarm Optimization (GPSO)

The emotion detection problem is 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. 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:

What is the search space and its dimension.

How do we represent a particle in the emotion-detection setting?

How do we represent the position and velocity of a particle?

What is the objective function to be minimized by the PSO.

In section II, 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, $q_1, q_2, \ldots, q_n$. Let $D_1, D_2, \ldots, D_n$ represent the domains of the AUs, $q_1, q_2, \ldots, q_n$ respectively. That is D_j represents a 2-dimensional rectangular neighborhood 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:

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

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

Definition 2: Particle, its 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 $(val_1, val_2, \ldots, val_n)$, where $val_j \in D_j$. Thus, $x_i(t)$ is a vector, $(val_1, val_2, \ldots, val_n)$.

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

There are two peculiar issues that make the emotion detection problem a little different from 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 position and its target. For example, the following is the definition of the objective function for the swarm that is targeting the happy emotion.

Definition 3: Objective function for the happy-targeting swarm:

Let $S = (s_1, s_2, \ldots, s_n)$ represent the happy emotion. Then the objective function for the happy swarm, $f_s : R^n \longrightarrow R$, is defined as:

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

The objective functions for the other swarms are defined similarly.

```
Create and initialize m swarms of n dimensions.
//m is the # of different emotions
//n is the # of action units being tracked.
Add a particle, Q, representing the positions of the AUs in
   each swarm
Read the approximate position of each emotion from a file.
For each frame of the video, do
For each swarm do
  for each particle in the swarm do
      //set the personal best position,  yi
      if f(xi) < f(yi) then
         yi = xi;
```

```
            end
            //set global best to be position of Q.
        end
        for each particle in the swarm do
            update the velocity using (7);
            update the position using (1);
        end
        compute the distance of the swarm from its target
    emotion
    end
        declare the target emotion of the swarm whose
    distance from
        its target is the shortest as the emotion being
    expressed in the
        current frame.
    end
```

Fig. 4.The GPSO algorithm.

Our proposal is that, in each iteration of the PSO algorithm, each swarm will update the positions of its particles as usual. These positions are then compared to find the swarm that is closest to its target. Such a swarm 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.

The second issue that makes the emotion-detection problem a little different from 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 $X_q(t) = (q_1, q_2, \ldots, q_n)$, where $q_1, q_2, \ldots, q_n$ 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 (2) to (7), where the position of Q is always regarded as the global best.

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

With these proposed changes, the particles are effectively 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. Fig. 4 summarizes the GPSO algorithm.

Experimental Results

The GPSO algorithm discussed in section III 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 approximate 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 click the relevant button to save the identified positions of the AUs into a file as the coordinate values for the particular emotion. The learning session is ended as soon as the data for each of 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. The GPSO algorithm is then executed to detect the emotions expressed in each frame of the video clip. The detected emotion is visually displayed on the screen.

Due to the modifications introduced 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 and identify the emotion being expressed.

RESULTS OF EMOTION DETECTION BY GPSO

Subject	Number of Frames	Success	% Success
Subject #1	600	476	79.3%
Subject #2	600	502	83.7%
Subject #3	600	559	93.2%
Subject #4	600	490	81.7%
Subject #5	600	472	78.7%
Subject #6	600	525	87.5%
Subject #7	600	551	91.8%
Subject #8	600	533	88.8%
Subject #9	600	478	79.7%
Subject #10	600	561	93.5%
Subject #11	600	486	81.0%
Subject #12	600	516	86.0%
Total	**7200**	**6149**	**85.4%**

For this study, we considered all the six universal basic emotions, namely happy, sad, surprise, disgust, anger and fear. These six, plus the neutral state gives seven possible states that the GPSO system can detect. We have tested the system with video clips recorded for 12 different subjects. On each subject, 3 sample video clips were recorded each of which contains 200 frames, giving a total of 600 frames per subject. In each video clip, subjects were asked to express the six different emotions randomly in the form, emotion X, neutral, emotion Y, neutral, etc.

In order to test the effectiveness of the detection system, we made the system to pause at each frame of the video clip so that in addition to the automatic detection, we also 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 were recorded in a file. Table I shows the success rates recorded for each subject, where a detection is determined to be successful if the auto-detection and the manual detection coincide.

From Table I, the success rates recorded ranges from 79.3% to 93.5%, with the average being 85.3%. 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, we observed that the errors were mainly found during transitions from neutral state to some emotion state or from some emotion state to the neutral state. In these transition states, it is really difficult to say exactly what the state is even to the human user.

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 improve the system so that the AUs are specified on the video clips rather than the subjects. The ultimate goal is to develop a real-time system that can be embedded it into a robot for some practical useful purposes.

Conclusions

We have presented a modification of the Particle Swam Optimization (PSO) algorithm that we designed for the purpose of emotion detection. The modified PSO algorithm, which we called, Guided Particle Swarm Optimization (GPSO), was implemented and tested on video clips that contain the six basic emotions. The results we obtained and presented in this paper show promising success rates. We noted that the algorithm was very efficient in terms of the speed with which particles converge to identify the emotion being expressed in each video frame. This is in part due to the concurrent nature of PSO 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. Another equally important factor contributing to the efficiency of the GPSO algorithm is the fact that it made particles to be guided by the actual positions of the AUs as the video clip is played. In future work, we shall improve the system to work with video clips in which subjects do not have AUs marked on their faces. We shall also study the run-time efficiency of the system compared to other methods.

ACKNOWLEDGMENT

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

REFERENCES

[1] J. Kennedy, R. Eberhart, "Particle swarm optimization", *Proceedings. of IEEE International Conference on Neural Networks*, Perth, Australia, 1995, pp. 1942-1948.

[2] K. P. Wang, L. Huang, C. G. Zhou, and W. Pang, "Particle Swarm Optimization for Travelling Salesman Problem", *Proceedings of International Conference on Machine Learning and Cybernetics*, 2003, pp. 1583-1585.

[3] V. Miranda and N. Fonseca, "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*, 2002, vol. 2, pp. 745-750.

[4] M. Settles and B. Rylander. , "Neural Network Learning using Particle Swarm Optimizers", *Advances in Information Science and Soft Computing*, 2002, pp. 224-226.

[5] M.G. Omran, A.P. Engelbrecht, and A. Salman, "Particle Swarm Optimization Method for Image Clustering", *International Journal on Pattern Recognition and Artificial Intelligence*, , 2005, vol. 19, no. 3, pp. 297-322.

[6] D.W. van der Merwe and A.P. Engelbrecht, "Data Clustering using Particle Swarm Optimization", *Proc. of IEEE Congress on Evolutionary Computation*, 2003, vol. 1, pp. 215-220.

[7] X. Xiao, E.R. Dow, R.C. Eberhart, Z. Ben Miled, and R.J. Oppelt, "Gene Clustering using Self-Organizing Maps and Particle Swarm Optimization", *Online Proceedings of the Second IEEE International Workshop on High Performance Computational Biology*, 2003.

[8] B.B. Thompson, R.J. Marks, M.A. El-Sharkawi, W.J. Fox, and R.T. Miyamoto, "Inversion of Neural Network Underwater Acoustic Model for Estimation of Bottom Parameters using Modified Particle Swarm Optimizer", in *Proc. of the International Joint Conference on Neural* Networks, 2003, pp. 1301-1306.

[9] Salman, I. Ahmad, and S. Al-Madani, "Particle Swarm Optimization for Task Assignment Problem", *Microprocessors and Microsystems*, 2002, vol. 26, no. 8, pp. 363-371.

[10] C.A. Coello Coello, E.H.N. Luna, and A.H.N. Aguirre, "Use of Particle Swarm Optimization to Design Combinational Logic Circuits", *Lecture Notes in Computer Science, Springer-Verlag* , 2003, vol. 2606, pp. 398-409.

[11] D. Keltner and P. Ekman, "Facial expression of emotion", *Handbook of emotions, M. Lewis and J.M. Haviland-Jones, Eds. Guilford Press, New York*, 2000, pp. 236-249.

[12] M. F. Valstar and M. Pantic. "Biologically vs. Logic Inspired Encoding of Facial Actions and Emotions in Video". *IEEE Inter. Conference on Multimedia and Expo*, 2006, pp. 325--328.

[13] M. Pantic and L.J.M. Rothkrantz, "Toward an Affect-Sensitive Multimodal Human-computer Interaction", *Proceedings of the IEEE*, 2003, vol. 91, no. 9, pp. 1370-1390.

[14] Y. Tian, T. Kanade and J.F. Cohn, "Recognizing action units for facial expression analysis", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2001, vol. 23, no. 2, pp. 97-115.

[15] M.S. Bartlett, G. Littlewort, C. Lainscsek, I. Fasel, J. Movellan, "Machine learning methods for fully automatic recognition of facial expressions and actions", *IEEE International Conference on Systems, Man and Cybernetics*, 2004, vol. 1, pp. 592-597

[16] M.F. Valstar, I. Patras and M. Pantic, "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*, 2005, vol. 3, pp. 76-84.

[17] Y. Zhang and Q. Ji. "Active and Dynamic Information Fusion for Facial Expression Understanding from Image Sequences", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, vol. 27, no. 5, pp. 699-714.

   a. P. Engelbrecht, "Computational Intelligence: An Introduction", Second Edition, New York, NY: Wiley, 2007.

[18] J. Kennedy, R. Mendes, "Population Structure and Particle Performance", in Proceedings of the *IEEE congress on Evolutionary Computation*, 2002, pp. 1671-1676.