A New Approach on Particle Swarm Optimization for Multimodal Functions

Zahra Afshahi
Information & Communication Technology Management
Systems & Quality V.P. MAPNA
Tehran, Iran
Electrical and Computer Engineering Department
Islamic Azad University
Qazvin, Iran
Afshahi_ai@yahoo.com

MohammadReza Meybodi
Computer Engineering and Information Technology
Department
Amirkabir University
Tehran, Iran
mmeybodi@ce.aut.ac.ir

Abstract—This paper describes a technique that extends PSO to handle multiple optima on a multimodal functions. In this paper, we present a new algorithm based on clustering particles to identify niches. For that we employ the standard k-means clustering algorithm which can identify the number of clusters adaptively. In each niche we used artificial immune system algorithm to determine the true members of it. Experimental results show that the proposed algorithm can successfully locate all optimum solutions on a small set of test functions during all simulation runs.

Keywords— Particle swarm optimization; Niche; Artificial immune system and k-means

I. INTRODUCTION

The particle swarm optimization is first proposed by Kennedy and Eberhart [1]. It is a stochastic optimization technique inspired by the behaviour of a flock of birds. This evolutionary computation technique has been shown to effectively solve unimodal optimization problems. PSO are however not well equipped to locating multiple optimal solutions, because the design of this algorithm is usually targeted to the goal of finding a single optimal solution for a given problem. In multimodal functions, PSO pick just one of the optimal solutions or it could even be misled by the presence of more than a single optimum and fail to converge.

For optimizing these function, PSO have been modified introducing the concept of niching when niching applied to PSO, it allows it to divide the space in different areas and search in parallel. In this paper we introduce a niching technique for PSO. Moreover, we will use the k-means clustering algorithm to identify niches in the population. After that, we use artificial immune system to identify the members of niche. In section 2, we present an overview of PSO and its improvements. Section 3 gives a brief overview of existing niching techniques in the fields of PSO optimizers. Section 4 is devoted to a detailed introduction of our new approach to PSO. In section 5, we show the results obtained by the algorithm from the experiments set up to compare its performance to those of other PSO niching algorithm.

II. ORIGINAL PSO

The original PSO was inspired by the social behaviour of birds flocking or fish schooling. This algorithm consists of a swarm of particles flying through the search space [1]. Each individual $i$ in the swarm contains parameters for position $X_i$, velocity $V_i$, and personal best position $y_i$, where $X_i \in \mathbb{R}^n$, $V_i \in \mathbb{R}^n$ and $y_i \in \mathbb{R}^n$ while $n$ is the dimension of the search space. The position of each particle represents a potential solution to the optimization. The personal best position associated with a particle's is the best position that particle has visited thus far, i.e. a position that yielded the highest fitness value for that particle. If $f$ denotes the objective function, then the personal best of a particle at a time step $t$ is updated as:

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

(1)

Depending on the social network structure of the swarm best and gbest experiences of particles, exchange among them. For the gbest model, the best particle is determined from the entire swarm. If the position of the best particle is denoted by the vector, then:

$$\hat{y}(t) = \left\{ y_0(t), y_1(t), ..., y_n(t) \right\}$$

$$\min \left( f(y_0(t)), f(y_1(t)), ..., f(y_n(t)) \right)$$

(2)

For the bbest model, a swarm is divided into overlapping neighborhoods of particles. For each neighborhood $N_j$, a best particle is determined with position $\hat{y}_j$. The best particle is referred to as the neighborhood best particle, defined as:

$$\hat{y}_j(t+1) = \min \left( f(y_j(t+1)), f(y_i(t+1)) \right)$$

(3)

In [2] Kennedy and Mendes recommended the Van-Neumann architecture, in which a particle's neighbors are above, below and on each side on a two dimensional lattice, to
be the most promising one. For each iteration of a gbest PSO algorithm, the jth dimension of particle $i$'s velocity vector, and its position vector, $X_i$, is updated as follows:

$$V_{ij}(t+1) = wV_{ij}(t) + C_1r_1(t)(x_{ij}(t) - x_{ij}(t)) + C_2r_2(t)(x_{ij}(t) - x_{ij}(t))$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1)$$

(4)

The PSO algorithm performs repeated applications of the update equations until a specified number of iterations have been exceeded, or until a user-defined stopping criterion has been reached.

III. NICHEING TECHNIQUES IN PSO

Niching techniques maintain multiple solutions in multimodal domains, in contrast to swarm intelligence optimization technique such as PSO that have been designed to only locate single solutions. Applying niching techniques in PSO algorithm extends the inherent unimodal nature of this algorithm by growing multiple swarms from an initial particle population. The initial particle swarm is split into smaller swarms as niches are detected. Upon termination of the algorithm, each sub swarm represents one of the potential solutions to the problems.

Thus as pointed out by Engelbrecht et al. the standard PSO must be modified to allow the efficient location of multiple solutions [9]. Notwithstanding the differences between the approaches we will discuss for the particle swarm, with respect to the evolutionary ones, we will still refer to them as niching technique.

A. Objective function stretching

The stretching technique adopts the landscape of an optimization problem's fitness function to remove local minima [8, 10]. This method was applied as a sequential niching technique to the PSO algorithm, by Parsopoulos and Vahatì [12]. When a local solution is detected during the evolutionary learning process, the stretching operator is applied to remove the detected solution from the fitness landscape. Subsequent iteration of the PSO algorithm can then focus on locating solutions in other parts of the search space, assured that the detected local optima will not again lead to premature convergence. "Application of the stretching functions" means that the fitness calculation of removing particles is adapted. In this way successive iterations of the search space, lead to the identification of the other solutions [11]. The effectiveness of the stretching transformation is not uniform on every function. In fact, in some cases it can introduce false minima, which render this method unreliable. Van den Bergh pointed that, keeping the PSO from discovering all possible solutions [25].

B. Niche PSO

In 2002, the nbest PSO introduced as a parallel niching. This method is aimed at locating multiple solutions to multimodal problems through the use of multiple, independent sub swarms [13]. A new approach, Niche PSO, was proposed which use sub swarms to locate multiple solutions to multimodal function optimization problems [14]. This algorithm starts by uniformly distributing particle in the search space of an optimization problem. The initial population of particles is referred to as the main swarm. It traverse the search space, they invariably no positions that have attractive fitness. A potential identified by monitoring the change in a particle's a number of training iteration. When such a identified a new sub swarm is created by removing main swarm the particle that detected the potential i creating a sub swarm from it. The algorithm is considered have converged when sub swarms no longer imp solutions that represent.

C. Species-based PSO

In this algorithm, a procedure determines the size of the niches in the population [16]. species seeds have been identified, all the others assigned to the niche formed by the closest seed neighborhood structure is adapted to reflect the niches. In fact, each species seed would serve as a species and all other particles in its niche. This method is suitable on dynamics environments [6]. How requires a radius parameter $\sigma$ to determine the niches.

D. Niching vector-based PSO

In this method, niches are identified by a sequential way [22]. This procedure starts from the first and then repeating for the particles outside its niche the same authors developed the parallel vector-based more efficient version, based on the same principle followed a parallel approach. In this method, niches are identified and maintained in parallel introduction of a special procedure which can make niches when they become closer than a specified The vector-base approach has the appealing identifying niches by using operations on vector inherent to the particle swarm algorithm. Thus, a good way to build a particles swarm for multimodal optimization.

E. Adaptive niching PSO

In this method the main parameters of niching adaptively [4]. The first step, the average distance between each its closest neighbor as:

$$r = \frac{\sum_{j=1}^{N} \min_{i \neq j} \|x_i - x_j\|}{N}$$

This parameter determines the formation of niches.

IV. OUR APPROACH

In this paper we proposed the clustering method for PSO to identify multiple global and local maxima in a multimodal search space. The basic idea is the inherent unimodal nature of the standard PSO growing multiple swarms from an initial population. The initial population (particle swarm) is split into swarms which called niche, to preserve diversity.
 niches in population we employ the standard k-means algorithm, which is probably the best known partition clustering algorithm. Passaro and Starita presented KPSO [3], which employed the standard k-means clustering algorithm and improved with a mechanism to adaptively identify the number of clusters, which is similar to ANPSO [4]. The main problem in these algorithms is determining the true numbers of each niche. In our approach, this problem is solved by using artificial immune system (AIS). In this section, we introduce our approach.

A. k-means

K-means is a very simple algorithm [24] and Kennedy used this algorithm to cluster particles in a swarm in his research on stereotyping [7]. Particles cluster according to their closest previous best position. After that the neighborhood topology modifies so that each particle can communicate only with particles in the swarm cluster. Therefore, the main swarm turns into a collection of sub swarms which tend to explore different regions of search space. Since, we want to perform a local search in each sub-swarm, each of them use a gbest topology, with all connected particles which are the members of that sub-swarm. The initial swarm of particles is randomly generated. Then, k-means algorithm is performed and repeated at regular intervals. Particles in different sub-swarm (cluster) at early stage of the simulation can end up in the same local optimum, means that those clusters are similar and merge by creating a new sub-swarm. Social information of them share between the other clusters, or in contrast when some particles of a cluster move towards a different optimum, that cluster can be split into clusters. In our method, clustering algorithm is applied only every C iteration. This idea maintains the clusters over time by blocking communication between particles in different clusters and allows particles to follow their natural dynamics for some steps. Moreover, the computational overhead becomes smaller than the other niching PSO technique.

Estimating the number of clusters

One of the major shortcomings of k-means algorithm is the number of clusters k [17]. In our approach, we optimize Bayesian Information Criterion (BIC) [15] to estimate the best value of k. We can calculate its BIC value with:

\[ \text{BIC}(D) = \ell(D) - \frac{D^2}{2} \log N \]  

(6)

This is also known as the Schwarz criterion. Where D is the number of clusters, \( \ell(D) \) is the log-likelihood point, and \( \rho \) is the number of parameters and is given by the sum of \( k^{-1} \) k-dimensional probability densities of \( k \) centroid coordinates and the k variance \( \sigma_i \), thus we have:

\[ \rho = (k-1) + d \cdot k + k \]

(7)

\[ \ell(D) = \frac{1}{N} \sum_{i=1}^{k} \left[ \rho - m_i \right] \]

where \( \sigma_i \) is the variance of the cluster \( C_i \), with \( m_i \) the mean of the \( j \)-th cluster, \( N_i \) its size. The formula for the log-

likelihood can be calculated considering that we are assuming components densities in the form of spherical Gaussians.

\[ \rho\left( x \left| m_i, \sigma_i \right. \right) = \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(x-m_i)^2}{2\sigma_i^2}} \]

(8)

And with a few mathematical transformations, the log-likelihood of the clustering \( \ell(D) \) can be written as:

\[ \ell(D) = \sum_{i=1}^{k} \log \rho(x_i | D) = \sum_{i=1}^{k} \ell(D_i) - N \cdot \log N \]

(9)

where, \( \ell(D_i) \) is the log-likelihood for each cluster \( D_i \):

\[ \ell(D_i) = -\frac{N_i}{2} \log 2\pi - \frac{N_i d}{2} \log \sigma_i^2 - \frac{N_i - 1}{2} \]

(10)

At each clustering application, k-means algorithm is thus repeated with varying value for k that usually in the range from 2 to \( \sqrt{N} \), then the clustering with highest BIC is chosen.

The other additional parameter is \( r_i \), the number of times for a single clustering application. Because of the depended of k-means algorithm on the initial assignment the seeds, repeat this algorithm a few times to obtain good results. Thus, we set \( r_i = 10 \).

C. Artificial immune system

After performing k-means algorithm and defining niches in population, we have to remove some particles from the overcrowded clusters. We use the concept of artificial immune systems. History and progression of research in the field of AIS, shows that works in this area has 3 major roots, and consequently distinct philosophies: idiothetic network theory, negative selection and danger theory [18, 19, 20, 21].

In this paper we use the theory of negative selection as a cutting procedure to avoid the formation of overcrowded niches. The negative selection algorithm (NSA) is the most widely used techniques in AISs [5]. The NSA is based on the principles of self-nonself discrimination. The algorithm was inspired by the thymic negative selection process that intrinsic to natural immune systems, consisting of screening and deleting self-reactive T-cells, i.e. these T-cells that recognize self-cells.

![Figure1. Simplified view of the thymic negative selection](image)

Resumption of this paper and in our approach negative selection responsible for eliminating all particles (T-cells
Labels) whose fitness is smaller than the average fitness of the niche which they belong it (receptors recognize and bind with self antigens represented in thymus). This concept is illustrated in Fig. 1.

In this problem the antigens are position vectors \( X(t) \). For indicating self and non-self antigen in this problem we need to calculate the average fitness of all the particles in each niche. These procedures continue by the following as in (8):

\[
\text{If } f(x_{i,j}) \leq f_{\text{average}} \Rightarrow x_{i,j} \text{ non-self} \\
\text{If } f(x_{i,j}) > f_{\text{average}} \Rightarrow x_{i,j} \text{ self}
\]

Where, \( x_{i,j} \) is the \( i_{th} \) particle in \( j_{th} \) niche. In a niche defines as an antigen remove from that niche and will add to a new niche, which known as \( N_{\text{explore}} \). At the end, all the members of \( N_{\text{explore}} \) reinitialize randomly. Using this procedure has two benefit, 1) avoiding the formation of overcrowded niches, which would end up in a waste of computational power, as too many particles would explore the search space around a single optimum, 2) reinitializing random all the members of \( N_{\text{explore}} \) cause to explore new areas. After performing AIS the members of each niche identify. Now we have \( k \) niches whose particles are fully connected, realizing a best topology in each niche. The particles in \( N_{\text{explore}} \) organize in a von Neumann lattice neighborhood, as it is shown in Fig. 2.

![Van-Neumann lattice neighborhood](image)

Fig. 2. Van-Neumann architecture for the particles in \( N_{\text{explore}} \)

Fig. 3 reports the pseudo code for main algorithm and AIS.

```
Procedure Main
Initialize particles with random positions and velocities.
Set particles pbests to their current positions.
Calculate particles fitness.
for step m=0 to T-1 do
   if t mod v=0 then
      Cluster Particles with K-means algorithm
      Execute the procedure Artificial Immune System
   end
   Update particles velocities.
   Update Particles positions.
   Recalculate particles fitness.
   Update particles and neighborhood best positions.
end.
```

Figure 3. Main algorithm pseudo code

V. EXPERIMENT

This section summarizes results of our approach to finding optimum of multimodal functions. We compare our algorithm with SPSO, ANPSO and KPSO algorithm. The study was conducted on the same set of benchmark functions which are reported in Table1. In order to show how our algorithm effectively identify niches surrounding the optimum function, we show in Fig. 4 several significant iteration algorithms on the Brain ROC function. Fig. 4 (a) shows how at the beginning of the run the particles of swarm are randomly distributed on the search space, and (c) they naturally start to split in different niches as shown in Fig. 4 (e), because of running AIS and some particles from overcrowded niches, at iteration each niche we have just 2 or 3 particles. Hence, it is shown in Table2 our method is the fast one among other. At the end at iteration, algorithm will eventually converge to exactly 3 niches or clusters become empty.

In Table2 we report the results obtained on benchmark functions with KPSO, SPSO and AP execution was repeated 30 times. Value of \( C \), the steps between two clustering applications, is a parameter that we have to define. With the higher \( C \), performance of the algorithm did not vary significantly.

VI. RESULT

In this paper we introduced a new approach niching technique in PSO algorithm, which allow it to optimize in multimodal functions. Then we described our approach used k-means algorithm for defining also AIS algorithm as a cutting procedure. The algorithm maintains essentially the structure of state. Moreover, in our approach niches can define a single cluster analysis, which showed comparable and higher performance entire test we conducted.

REFERENCES


Table 1. list of test functions

<table>
<thead>
<tr>
<th>Name</th>
<th>F</th>
<th>Global Optimums</th>
<th>Search Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Branin ROC</td>
<td>$$f(x, y) = \left( y - \frac{5x^2}{4\pi^2} + \frac{5x}{\pi} - 6 \right)^2 + 10\left(1 - \frac{1}{8\pi}\cos(x)+10\right)$$</td>
<td>3</td>
<td>$$-5 \leq x \leq 10$$, $$0 \leq y \leq 15$$</td>
</tr>
<tr>
<td>Six-Hump camel back</td>
<td>$$f(x, y) = -\left(4 - 2.1x^2 + \frac{x^4}{3}\right)\left(1 + xy + (-4 + \frac{4y^2})y^2\right)$$</td>
<td>2</td>
<td>$$-1.25 \leq x \leq 1$$, $$-1.2 \leq y \leq 1$$</td>
</tr>
<tr>
<td>Himmelblau</td>
<td>$$f(x, y) = 200 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2$$</td>
<td>4</td>
<td>$$-6 \leq x, y \leq 6$$</td>
</tr>
</tbody>
</table>

Table 2. Number of evaluations required to find all global optima

<table>
<thead>
<tr>
<th>Function</th>
<th>Numbers of particles</th>
<th>SPSO</th>
<th>ANPSO</th>
<th>k-PSO</th>
<th>k-PSO with AIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$$f_1$$</td>
<td>30</td>
<td>3169±0.92</td>
<td>3323±5220</td>
<td>2084±440</td>
<td>1854±471</td>
</tr>
<tr>
<td>$$f_2$$</td>
<td>30</td>
<td>2872±0.027</td>
<td>2798±857</td>
<td>1124±216</td>
<td>986±584</td>
</tr>
<tr>
<td>$$f_3$$</td>
<td>30</td>
<td>4096±0.018</td>
<td>16308±13157</td>
<td>2259±538</td>
<td>1789±603</td>
</tr>
</tbody>
</table>

Figure 4. Significant iterations of our approach: (a) iteration 10, (b) iteration 500, (c) iteration 1000, (d) iteration 1500, (e) iteration 1700, (f) iteration 2000, (g) iteration 2100.