A New Approach on Particle Swarm Optimization for Multimodal Functions

Zahra Afsahi Mohmmad Reza Meybodi

Afsahi_AI@yahoo.com Meybodi@ce.aut.ac.ir

Abstract: This paper describes a technique that extends PSO to locate 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.

1. Introduction

The particle swarm optimization is first proposed by Kennedy and Eberhart. It is a stochastic optimization technique inspired by the behavior of a flock of birds. This evolutionary computation technique [3][18][28] 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 [23]. 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, allows it to divide the space in different areas and search them in parallel. In this paper we introduce a niching technique in the PSO.

Moreover, we will use the k-means clustering algorithm to identify niches in the population [4]. After that, we propose artificial immune system to identify the members of each niche. In section 2, we present an overview of PSO algorithm. Section 3 gives a brief overview of existing niching techniques in the fields of PSO optimizers. Section 4 is dedicated to a detailed introduction of our new approach to niching PSO. In section 5, we show the results obtained by the new algorithm from the experiments set up to compare its performance to those of other PSO niching algorithm.

2. Original PSO

The original PSO was inspired by the social behavior of birds flocking or fish schooling [1]. This algorithm consists of a swarm of particles flying through the search space. Each individual *i* in the swarm contains parameters for position x_i, velocity v_i, and personal best position y_i , where $x_i \in R^n$,$v_i \in R^n$, $y_i \in R^n$ 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 *i* 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:

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

Depending on the social network structure of the swarm *lbest* and *gbest* experience of particles, exchange among them [3]. 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 $\hat{y}$, then

$$\hat{y}(t) \in \{y_0, y_1, \dots, y_s | f(\hat{y}(t))\} = min\{f(y_0(t)), f(y_1(t)), \dots, f(y_s(t))\} \quad (2)$$

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

$$\left\{\hat{y}_j(t+1) \in N_j \middle| f\left(\hat{y}_j(t+1)\right)\right\} = \min\{f(y_i)\} \quad \forall y_i \in N_j \ (3)$$

In [12], 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 j_{th} dimension of particle i_{th}'s velocity vector, and it's position vector,x_i, is updated as follows:

$$V_{i,j}(t+1) = w * V_{i,j}(t) + C_1 r_{1,j}(t)\left(y_{i,j}(t) - x_{i,j}(t)\right) + C_1 r_{1,j}(t)\left(\hat{y}_j(t) - x_{i,j}(t)\right) \quad (4)$$

$$x_i(t+1) = x_i(t) + V_i(t+1) \quad (5)$$

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.

3. Nihing technique 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 [8]. 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 [9].

Thus as pointed out by Engelbrecht et al. the standard PSO must be modified to allow the efficient location of multiple solutions. Notwithstanding the differences between the approaches we will discuss for the particle swarm, with respect to the evolutionary ones [30][29][27][26][7], we will still refer to them as niching technique.

3.1 Objective function stretching

This method was applied as a sequential niching technique [10] by Parsopoulos and Vahatis. The stretching technique adopts the landscape of an optimization problem's fitness function to remove local minimum [15][16]. 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. 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 [17].

3.2 Niche PSO

Niche PSO is aimed at locating multiple solutions to multimodal problems through the use of multiple, independent sub swarms [31][24]. This algorithm starts by uniformly distributing particle throughout the search space of an optimization problem. The initial swarm of particles is referred to as the main swarm. As particles traverse the search space [32], they invariably move towards positions that have attractive fitness. A potential solution is identified by monitoring the change in a particle’s fitness over a number of training iteration. When such a solution is identified a new sub swarm is created by removing from the main swarm the particle that detected the potential solution and creating a sub swarm from it. The algorithm is considered to have converged when sub swarms no longer improve on the solutions that represent [33].

3.3 Species-based PSO

In this algorithm, a procedure determines the species seeds, which identify the niches in the population [9]. Once the species seeds have been identified, all the other particles are assigned to the niche formed by the closest seed, and the neighborhood structure is adapted to reflect the division in niches. In fact, each species seed would serve as the *n*best for all the other particles in its niche. This method proved to be suitable on dynamics environments. However, it still requires a radius parameter σ to determine the extension of the niches [35].

3.4 Niching vector-based PSO

In this method, niches are identified by original in a sequential way [34]. This procedure starts from the global best and then repeating for the particles outside its niche. In this case, the different niches are identified and maintained in parallel, with the introduction of a special procedure which can merge two niches when they become closer than a specified threshold ε. The vector-base approach has the appealing property of identifying niches by using operations on vectors which are inherent to the particle swarm algorithm. Thus it provides a good way to build a particles swarm for multimodal function optimization [5].

3.5 adaptive niching PSO

In this method the main parameters of niching determine adaptively.

The first step, the average distance between each particle and its closest neighbor as

$$r = \frac{\sum_{i=1}^{N} min_{j\neq i}\left\|x_i - x_j\right\|}{N} \quad (6)$$

This parameter determines the formation of niches [25].

4. Our approach

In this paper we proposed the clustering based niching method for PSO to identify multiple global and local optima in a multimodal search space. The basic idea is to apply the biological concept of species in separate ecological niches to PSO to preserve diversity. To model species we use a multi swarm approach, one population for each species. To identify species in population of particles we employ the standard *k-means* algorithm, which is probably the best known partition clustering algorithm.

4.1 k-means

K-mean is a very simple algorithm [22] and Kennedy used this algorithm to cluster particles in a swarm in his research on stereotyping [21]. Particles cluster according to their *pbest,* the previous best position. After that the neighborhood topology modify so that each particle can communicate only with particles in the swarm cluster [4]. Therefore, the main swarm turns in a collection of sub swarms which tend to explore different regions of search space.

```
Procedure Main
  Initialize particles with random positions and velocities.
    Set particles' pbests to their current positions.
    Calculate particles' fitness.
    for step t=0 → T-1 do
        if t mod c=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
end.
```

Fig1. Main algorithm pseudo code

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 swarms. The initial swarm of particles is randomly generated. Then k-means algorithm is performed and repeated at regular intervals [11]. Particles in different sub swarm (cluster) at early stage of the simulation can end up in the same local optimum, it 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 fly towards a different optimum, that cluster can be split into too 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.

4.2 Estimating the number of clusters

One of the major shortcomings of k-means algorithm is the number of cluster k [22]. in our approach we optimize Bayesian Information criterion (BIC) to estimate the best choice of k. we can calculate its BIC value with

$$BIC(D) = \ell(D) - \frac{\rho}{2}.\log N \quad (7)$$

This is also known as the Schwarz criterion. Where D is a given clustering and $\ell(D)$ is the log-likelihood point, and ρ is the number of parameters and is given by the sum of *k-1* class probabilities $d.k$ centroid coordinates and the k variance estimate σ_j , thus we have

$$\rho = (k-1) + d.k + k \quad (8)$$

$$\sigma_j^2 = \frac{1}{N_j - 1} \sum_{p \in C_j} \left\| p - m_j \right\|^2 \quad (9)$$

Where σ_j is the variance of the clusterC_j, with m_j the center of the j_{th} cluster, N_j its size. The formula for the log-likelihood can be calculated considering that we are assuming components densities in the form of Spherical Gaussians.

$$\rho\left(x \middle| m_j, \sigma_j\right) = \frac{1}{\sigma_j^d \sqrt{2\pi}} e^{-\left(1 \middle| 2\sigma_j^2\right)} \left\| x - m_j \right\|^2 \quad (10)$$

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

$$\ell(D) = \sum_{i=1}^{N} \log \rho(x_i|D)$$
$$= \sum_{j=1}^{k} \ell(D_j) - N.\log N \quad (11)$$

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

$$\ell(D_j) = -\frac{N_j}{2}.\log 2\pi - \frac{N_j.d}{2}.\log \sigma_j^2 - \frac{N_j - 1}{2} + N_j.\log N_j \quad (12)$$

at each clustering application, k-means algorithm is thus repeated with varying value for k that usually in the range from 2 to $N/_2$, then the clustering with highest BIC is chosen [6]. The other additional parameter is r_k, the number of times for a single clustering application. Because of the depended of k-mean algorithm on the initial assignment the seeds, repeat this algorithm a few times to obtain good results. Thus we set $r_k = 10$.

4.3 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* [14]. History and progression of research in the field of AIS, shows that works in this area has 3 major roots, and consequently distinct philosophies: idiotypic network theory, negative selection and danger theory [19]. In this paper we use *the theory of negative selection* as a cutting procedure to avoid the formation of overcrowded niches. The negative selection is drawn from this question:" How does the immune system behave when it is confronted with a self antigen?" the answer to this question is rather complex, controversial and involves different mechanisms for B-cells and T-cells. This process responsible for eliminating all T-cells whose receptors recognize and bind with self antigens represented in thymus. The immune system with its stochastic cell production produce more cells to be able to get out of thymus [20].

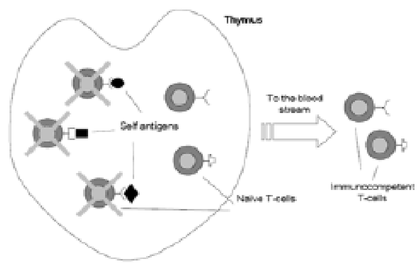


Fig2. Simplified view of the thymic negative selection

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 equation.

If $f(x_{i,j}) \leq f_{avrage} \;\; \Rightarrow \;\; x_i \;\; non-self$

If $f(x_{i,j}) < \; f_{avrage} \;\; \Rightarrow \;\; x_i \;\; self$

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 $N_{explore}$. At the end, all the members of $N_{explore}$ 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 randomly all the members of $N_{explore}$ 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 *gbest* topology in each niche. The particles in $N_{explore}$ organize in a von Neumann lattice neighborhood [2]. Figure 1, 3 reports the pseudo code for main algorithm and AIS.

Procedure Artificial Immune System
  for each cluster C_j **do**
    Calculate average Fitness of particles, F_{avg}
    for each particle **do**
      if $F(x_{i,j}) > F_{ave}$ then
        $x_{i,j} \rightarrow non-self$
        Adapt the neighborhood structure for the particles in C_j .
      else
        $x_{i,j} \rightarrow self$
        Remove the Particles from C_j
        Add the particles to M
      end
    End
    Reinitialize the M un-niched particles.
  end
end.

Fig3. AIS algorithm pseudo code

5. 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 can effectively identify niches surrounding the optima of a function, we show in figure 5 several significant iterations of our algorithm on the Branin ROC function. Figure5 (a) is shown how at the beginning of the run the particles of the main swarm are randomly distributed on the search space and in (b), (c) they naturally start to split in different niches. As it is shown in figure6, because of running AIS and removing some particles from overcrowded niches, at iteration 1700, in each niche we have just 2 or 3 particles. Hence, just as it is shown in table2 our method is the fast one among 3 other algorithm. It is interesting to note that, when the fist clustering algorithm, k-means, performs it defines 5 niches (figure6) of which actually 3 of them correspond to the global optima, while the other is a spurious one. At the end at iteration 1854 the algorithm will eventually converge to exactly 3 niches and as it shown in (figure 6) two clusters become empty.

Table1. List of test functions

Name	*F*	*Global Optimums*	*Search Range*
Branin ROC		3	$-5 \le x \le 10$ $0 \le y \le 15$
Six-Hump camel back		2	$-1.9 \le x \le 1.9$ $-1.1 \le y \le 1.1$
Himmelblau		4	$-6 \le x , y \le 6$

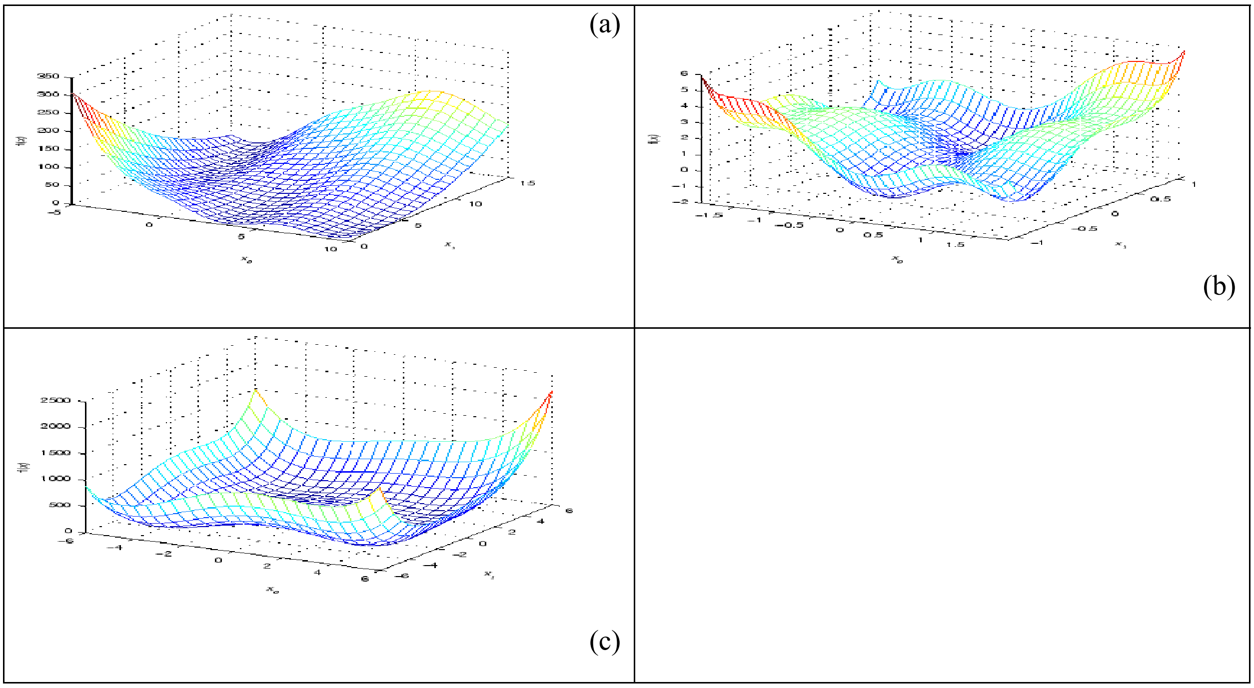


Fig4. Multimodal Function: (a) F1, (b) F2 and (c) F3.

Table 1: Number of evaluations required to find all global optima

Function	Numbers of particles	SPSO	ANPSO	k-PSO	k-PSO with AIS
F_1	30	3169±692	3323 ± 5220	2084±440	1854±471
F_2	30	2872±827	2798±857	1124±216	986±284
F_3	30	4096±2018	16308±13157	2259±538	1789 ± 603

In table2 we report the results obtained on the 3 benchmark functions with KPSO, SPSO and APSO. Each execution was repeated 30 times. Value of C, the number of steps between two clustering applications, is another parameter that we have to define. With the higher values, the performance of the algorithm did not vary significantly.

6. Result

In this paper we introduced a new approach to using niching technique in PSO algorithm, which allow it to find all optima in multimodal functions. Then we described that how our approach used k-means algorithm for defining niches and also AIS as a cutting procedure. The resulting algorithm maintains essentially the structure of standard PSO. Moreover, in our approach niches can define adaptively. Result showed comparable and higher performance in the entire test we conducted.

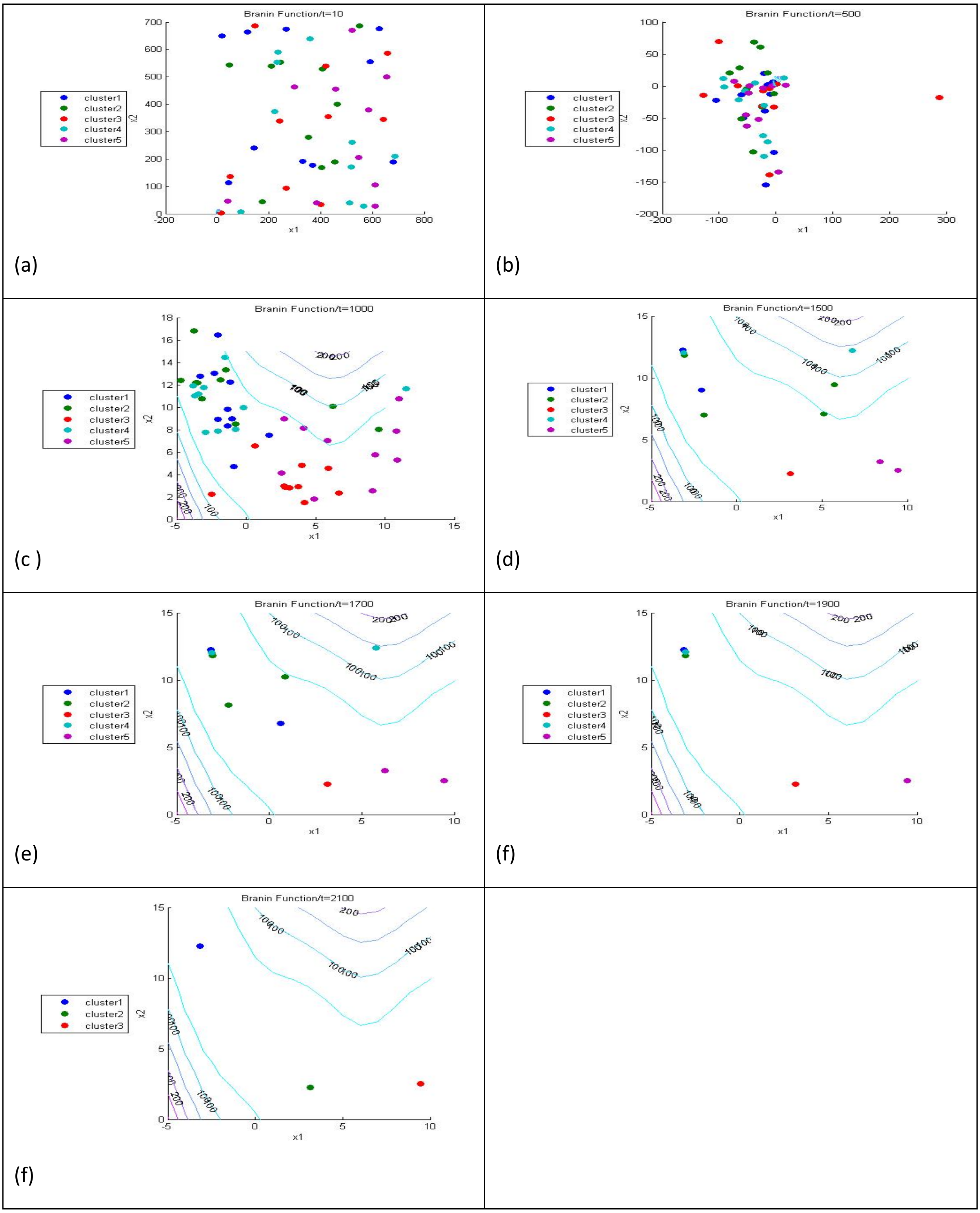
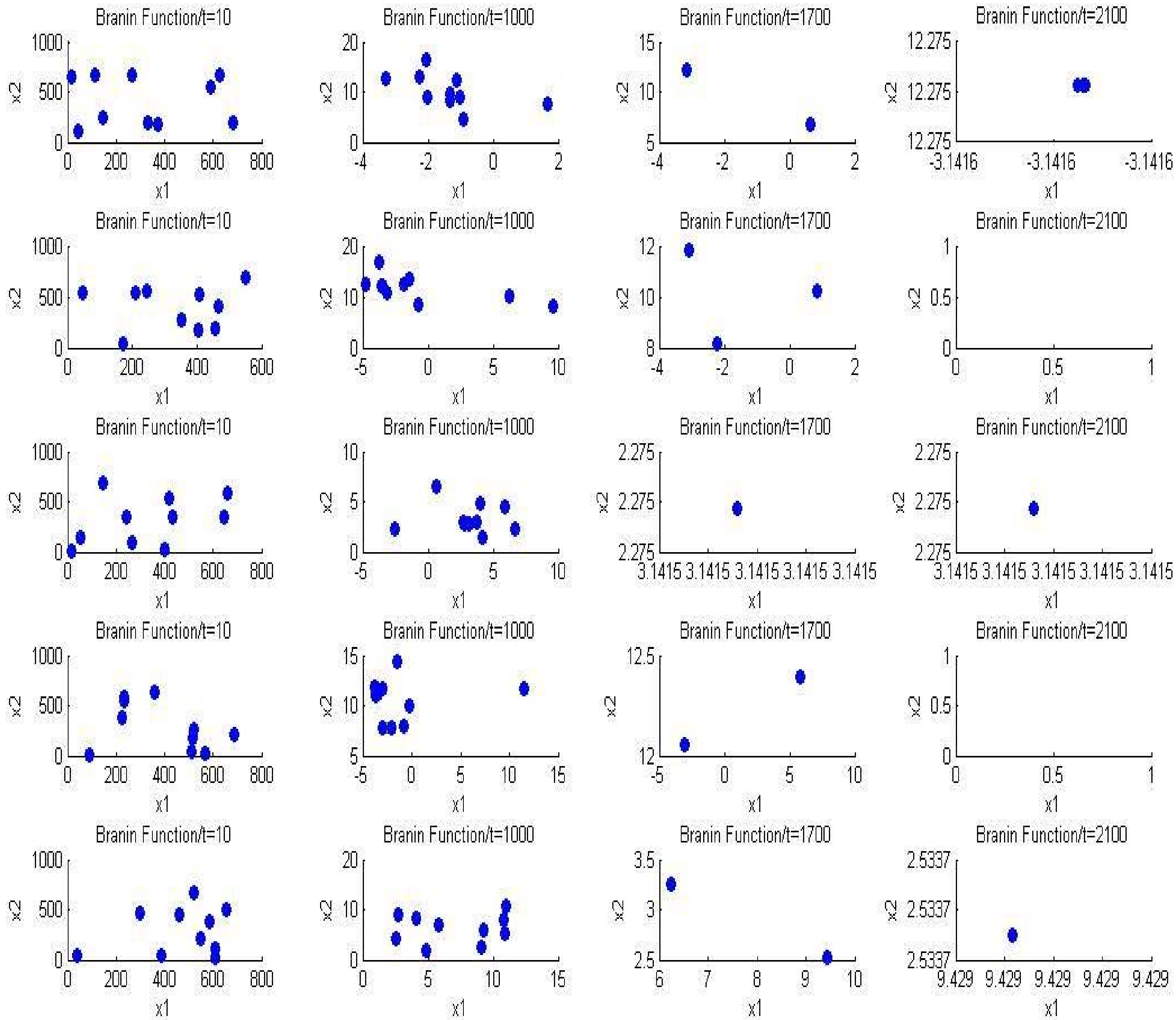


Fig5. Significant iterations of our approach: (a) iteration 10, (b) iteration 500, (c) iteration 1000, (d) iteration 1500, (e)iteration 1700, (f)iteration 1900, (g)iteration 2100


Branin Function/t=10
Branin Function/t=1000
Branin Function/t=1700
Branin Function/t=2100
x1
x2

7. References

[1] J. Kennedy and R. C. Eberhart, ”Particle swarm optimization”, in Proceedings of IEEE International Conference on Neural Networks(ICNN ‘95), vol. 4, pp. 1942-1948, IEEE Service Center, Perth, Western Australia, November-December 1995.

[2] J. Kennedy, S. Worlds and M. Minds, “Effects of Neighborhood Topology on Particle Swarm Performance”, Proceeding of the IEEE Congress on Evolutionary Computation, Vol. 3, pp. 1931-1938, July 1999.

[3] S. W. Mahfoud, “Niching method for genetic algorithms”, PHD thesis, University of Illinois at Urbana-Champaign, Champaign, Ill, USA, 1995.

[4] A. Passaro and A.Starita,” Particle swarm optimization for multimodal functions: a clustering approach”, Journal of the Artificial Evolution and Applications, vol. 2008, no.482032, February 2008.

[5] I. L. Schoeman and A. P. Engelbrecht, “A parallel vector-based particle swarm optimizer,” in Proceedings of the 7th International Conference on Artificial Neural Networks and Genetic Algorithms (ICANNGA ‘05), Coimbra, Portugal, March 2005.

[6] D. Pelleg and A. Moore, “X-means: extending k -means with efficient estimation of the number of clusters”, in Proceedings of the 17th International Conference on Machine Learning (ICML ‘00), pp. 727-734, Morgan Kaufmann, Stanford, Calif, USA, June-July 2000.

[7] J. Kennedy, “Stereotyping: improving particle swarm performance with cluster analysis”, in Proceedings of IEEE Congress on Evolutionary Computation (CEC’00), vol. 2, pp. 1507-1512, La Jolla, Calif, USA, July 2000.

[8] R. Brits, A. P. Engelbrecht, and F. van den Bergh, “Scalability of niche PSO”, in Proceedings of the IEEE Swarm Intelligence Symposium (SIS ‘03), pp. 228-234, Indianapolis, Ind, USA, April 2003.

[9] X. Li, “Adaptively choosing neighborhood bests using species in a particle swarm for multimodal function optimization”, in Proceedings of the Conference on Genetic and Evolutionary Computation(GECCO ’04), K. Deb, R. Poli, W. Banzhaf, et al., Eds., vol. 3102 of Lecture Notes in Computer Science, pp. 105-116, Springer, Seattle, Wash, USA, June 2004.

[10] D. Beasley, D. R. Bull, and R. R. Martin, “A sequential niche technique for multimodal function optimization”, Evolutionary Computation, vol. 1, no. 2, pp. 101-125, 1993

[11] S. H. Zahiri and S. A. Seyedin, “Swarm intelligence based classifiers”, Journal of the Franklin Institute, vol. 344, pp. 362–376, 2007

[12] Y. Zheng, L. Ma, L. Zhang and Qian, “Empirical study of particle swarm optimizer with increasing inertia weight”, In Proceeding of IEEE Congress on Evolutionary Computation, pp. 221-226, 2003

[13] K. C. Tan, C. K. Goh, A. A. Mamun, and E. Z. Ei, “An evolutionary artificial immune system for multi-objective optimization”, European Journal of Operation Research, vol. 187, pp. 371-392, June 2008.

[14] S. A. Hofmeyr, and S. Forrest, “Immunity by design: An artificial immune system”, In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO), vol. 2, pp. 1289-1296, 1999.

[15] F. V. Bergh, “An analysis of particle swarms optimizers”, Survey, November 2001.

[16] K. E. Parsopoulos and M. N. Vrahatis, “Modification of the particle swarm

optimizer for locating all the global minima", In Artificial Neural Network and Genetic Algorithms, Computer Science, pp. 324-327, Springer, Wien, Austria, 2001.

[17] K. E. Parsopoulos, V. P. Plagianakos, G. D. Magoulas, and M. N. Vrahatis, "Stretching technique for obtaining global minimizes through particle swarm optimization", in Proceedings of the Particle Swarm Optimization Workshop, pp. 22-29, Indianapolis, Ind, USA, April 2001.

[18] A. P. Engelbrecht, B. S. Masiye, and G. Pampara, "Niching ability of basic particle swarm optimization algorithms", in Proceedings of the IEEE Swarm Intelligence Symposium (SIS '05), pp. 397-400, Pasadena, Calif, USA, June 2005.

[19] P. K. Harmer, P. D. Williams, G. H. Gunsch, G. B. Lomont, "An artificial immune system architecture for computer security applications", IEEE Transactions on Evolutionary Computation, vol. 6, pp. 252-280, June 2002.

[20] C. A. Coello, and N. C. Cortes, "Solving multi-objective optimization problems using an artificial immune system", Kluwer Academic Publishers, Mexico, no. 2508, February 2002

[21] A. Passaro, and A. Starita, "Clustering particles for multimodal function optimization", in Proceeding of ECA Workshop on Evolutionary Computation, Riva del Garda, Italy, August 2006.

[22] R. Xu and D. Wunsch II, "Survey of clustering algorithms", IEEE Transactions on Neural Networks, vol. 3, no. 16, pp. 645-678, 2005

[23] P. J. Angeline, "Evolutionary optimization versus particle swarm optimization: philosophy and Performance differences", in proceedings of the 7th International Conference on Evolutionary Programming (EP '98), pp. 601-610, Springer, San Diego, Calif, USA, March 1998.

[24] R. Brits, A. P. Engelbrecht, and F. van den Bergh, "A niching particle swarm optimization", *in* Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL'02), vol. 2, pp. 692-696, Singapore, November 2002.

[25] S. Bird and X. Li, "Adaptively choosing niching parameters in a PSO," in Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation (GECCO '06), vol. 1, pp. 3–10,ACM Press, Seattle, Wash, USA, July 2006.

[26] D. E. Goldberg, J. Richardson, "Genetic algorithm with sharing for multimodal function optimization", in Proceedings of the second International Conference on Genetic Algorithms, pp. 41-49, 1987.

[27] J. Horn, "The nature of niching: genetic algorithms and the evolution of optimal, cooperative population", Doctoral dissertation, 95001, Urbana, University of Illinois, Illinois Genetic algorithm Lab, 1997.

[28] S. W. Mahfoud, "Niching methods for genetic algorithms", IlliGAL Rep. 95001, Urbana, University of Illinois, Illinois Genetic algorithm Lab, 1995.

[29] K. A. de Jong, "An analysis of the behavior of a class of genetic adaptive systems", PhD thesis, Dept. of Computer and Communication Sciences, University of Michigan, 1975.

[30] G. Harik, "Finding multiple solutions using restricted tournament selection, in L. J. Eshelman(Ed.), Proceedings of the sixth international Conference on Genetic Algorithms, pp. 24-31, San Francisco: Morgan Kaufman, 1995.

[31] R. Brits, A. P. Engelbercht, and F. van den Bergh, "Solving systems of unconstrained equations using particle swarm

optimization", in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '02), vol. 3, pp. 100-105, Hammamet, Tunisia, October 2002.

[32] R. Brits, A. P. Engelbercht, and F. van den Bergh, "A niching particle swarm optimization", in Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAl '02), vol. 2, pp. 692-696, Singapore, November 2002.

[33] A. P. Engelbercht and F. van den Bergh, "A new locally convergent particle swarm optimizer", in Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC '02), vol. 3, pp. 96-101, Hammamet, Tunisia, October 2002.

[34] I. L. Schoeman and A. P. Engelbrecht, "Using vector operations to identify niches for particle swarm optimization," in Proceedings of the IEEE International Conference on Cybernetics and Intelligent Systems (CCIS '04), vol. 1, pp. 361-366, Singapore, December 2004.

[35] J. P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization", Evolutionary Computation, vol. 10, no. 3, pp. 207-234, 2002. J. R. Beveridge", segmenting images using localizing histograms and region merging", Int.J.of Compt. Vision. vol. 2, 1989.