Int J Adv Manuf Technol (2012) 60:723–732
DOI 10.1007/s00170-011-3632-8

ORIGINAL ARTICLE

A novel algorithm based on hybridization of artificial immune system and simulated annealing for clustering problem

Khabat Abdi · Mohammad Fathian · Ehram Safari

Received: 15 February 2010 / Accepted: 9 September 2011 / Published online: 18 October 2011

Abstract A hybrid clustering method is proposed in this paper based on artificial immune system and simulated annealing. An integration of simulated annealing and immunity-based algorithm, combining the merits of both these approaches, is used for developing an efficient clustering method. Tuning the parameters of method is investigated using Taguchi method in order to select the optimum levels of parameters. Proposed method is implemented and tested on three real datasets. In addition, its performance is compared with other well-known meta-heuristics methods, such as ant colony optimization, genetic algorithm, simulated annealing, Tabu search, honey-bee mating optimization, and artificial immune system. Computational simulations show very encouraging results in terms of the quality of solution found, the average number of function evaluations and the processing time required, comparing with mentioned methods.

Keywords Clustering · Meta-heuristic · Artificial immune system · Simulated annealing

1 Introduction

Data mining is a process to extract knowledge from data. This could be possible discovering patterns from data associated with prior behavior of processes [1]. Nowadays, increase in amount of organization data and highly competitive environment of market, data mining enjoys a great popularity. Classification, estimation, prediction, association rules, and clustering are fundamental tools of data mining. Clustering is to group similar objects in a data set. An important issue in clustering is to distribute N objects to K clusters in a way that data of a cluster are similar to each other and have the most dissimilarity with those of other clusters [1]. Similarity and dissimilarity are defined based on selected metric distance criteria. Comparing with classification it is to say that clustering is an unsupervised classification in which classes are not specified formerly. Clustering can create a view as an independent tool in data distributing; moreover as a step of a pre-process can be served to other algorithms. Clustering has several applications such as text recognition, spatial data processing, image processing, market segmentation, etc.

In general, basic approaches of clustering consist of: partitional, hierarchical, density-based, space-gridding-based, model-based, and fuzzy clustering. Target functionalities of clustering problems are usually nonlinear and unconvex. Therefore, some clustering algorithms may fall into local optimum. In addition, they put a final complexity derived from number of clusters.

The methods, namely, meta-heuristics are a computational technique that optimizes a problem by iteratively trying to improve a candidate solution in connection with a given measure of quality. Algorithms such as genetic algorithm (GA), simulated annealing (SA), tabu search (TS), ant colony (ACO), artificial immune system (AIS), and etc. are meta-heuristic methods to optimize real problems [2–5].

Aforementioned factors make clustering problem as a NP-hard problem [6]. Consequently, there are several numbers of meta-heuristic algorithms which are proposed in order to resolve clustering problems as ACO [6], TS [7], GA [8], SA [9], honey-bee mating (HBMO) [10], etc.

K. Abdi (✉) · M. Fathian · E. Safari
Departments of Industrial Engineering,
Iran University of Science and Technology,
Narmak,
Tehran 16846, Iran
e-mail: khabatabdi@gmail.com

One of the meta-heuristic varieties is AIS-based algorithms inspiring from the way in which body immune system response to foreign factors [11]. These algorithms are exploited to solve a great verity of optimization problems [5, 12–16].

In this paper, for improving clustering precision, we provide an effective meta-heuristic algorithm based on hybridization of SA and AIS, namely, immune system-simulated annealing (IS-SA).

The organization of this paper is as following. In Section 2, the AIS concepts and theories are described. The proposed hybrid algorithm to solve clustering problem using AIS and SA is presented in Section 3. In Section 4 parameters of proposed algorithm are tuned using Taguchi method, then results and performance of IS-SA is compared with those of conventional meta-heuristic algorithms. Finally, conclusion and future works are reported in Section 5.

2 Artificial immune system

Artificial immune system is defined in [17] as follow: "Adaptive systems inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving".

There are several reasons behind the idea of using AIS as an inspiration source to design algorithms. They could be found in AIS properties consisting self-organization, learning and memory, adaptation, recognition, robust, and scalability [11]. The aim of the immune system is to protect the human body from foreign invaders. These foreign invaders are recognized as anti-genes and an immune response is impressed by stimulating antibody associating with that antigen. Immune response illustrates how antibodies understand structure of pattern of antigens and destroy them. Immune systems basically consist of lymphocytes which are white cells of blood and B and T cells. These cells help in recognition and destroying invaders process. AI has diversity of application such as detection intrusion [18], network security [19], data analysis [12], etc. There are several algorithms in AIS which can be described as follows:

(a) *Clonal selection theory*: Here in this theory, it is proposed that only that part of cell enjoying the capability of antigen detection would be cloned. Clonal selection applies on both of T and B cells [20]. There are several types of clonal selection algorithms, most of them are used in optimization problems. Some examples of mentioned types are opt-aiNet [21], B-cell algorithm [22], CLONALG [23]. The purpose of such algorithms is to improve candidate solution in algorithm iterations which is performed by cloning, mutation, and selection. Such application makes clonal selection similar in functionality to genetic algorithm. In this paper, opt-aiNet is selected for basic immune method. Summary of this method is the following: The opt-aiNet algorithm was first presented in 2002 [21]. One of the two authors of opt-aiNet, Timmis, is also the author of the B-cell algorithm. In spite of the fact that opt-aiNet was published earlier than B-cell algorithm (was published in 2003 [22]). The applied parameters in opt-aiNet algorithm can be listed as: $|P|$—population size, c—number of clones in the clonal pool, t_{ei}—average error improvement threshold which is responsible for execution of the network interactions, t_{aff}—affinity threshold, i.e., the minimal distance among two antibodies, the antibodies placed closer to each other than this distance are marked as being too close and one of them should undergo suppression, r_{nc}—newcomers ratio which is used to calculate the number of new antibodies added to the population. The number is calculated by multiplying r_{nc} by the number of antibodies in the population remaining after the step of elimination of antibodies being too close to each other.

(b) *Negative selection theory*: The main idea of this algorithm is that body immune system has the capability of recognition between invaders and factors come from host cells. In [24], an algorithm is presented based on this theorem.

(c) *Immune network theory:* The basic idea behind this theory is to reach to immune memory through a mutual amplifying network between B cells [25]. One of the popular algorithms based on this theorem is aiNet [26]. In this algorithm, members which they have, matching less than a threshold with training data items, would be removed.

3 IS-SA algorithm

The IS-SA algorithm is based on simulated annealing and immune-based methods in which the solution is obtained through employing iterations of cloning, mutation, and enrichment operators as well as considering interactions among antibodies. The enrichment operator is a novel operator developed by authors to be specifically applied in clustering problem where it is performed on an antibody to improve its quality. Additionally, Boltzmann criterion [27] has been used for acceptance of potential solutions.

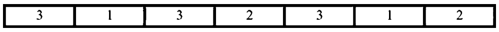
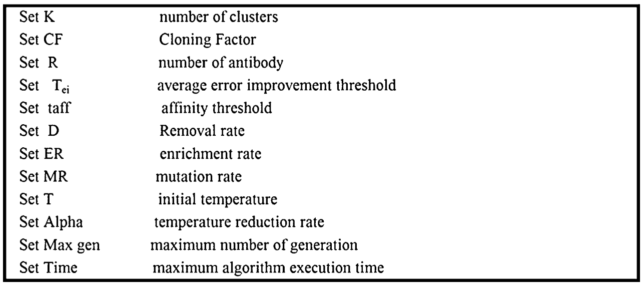


Fig. 1 String representation



Set K	number of clusters
Set CF	Cloning Factor
Set R	number of antibody
Set T_{ei}	average error improvement threshold
Set taff	affinity threshold
Set D	Removal rate
Set ER	enrichment rate
Set MR	mutation rate
Set T	initial temperature
Set Alpha	temperature reduction rate
Set Max gen	maximum number of generation
Set Time	maximum algorithm execution time

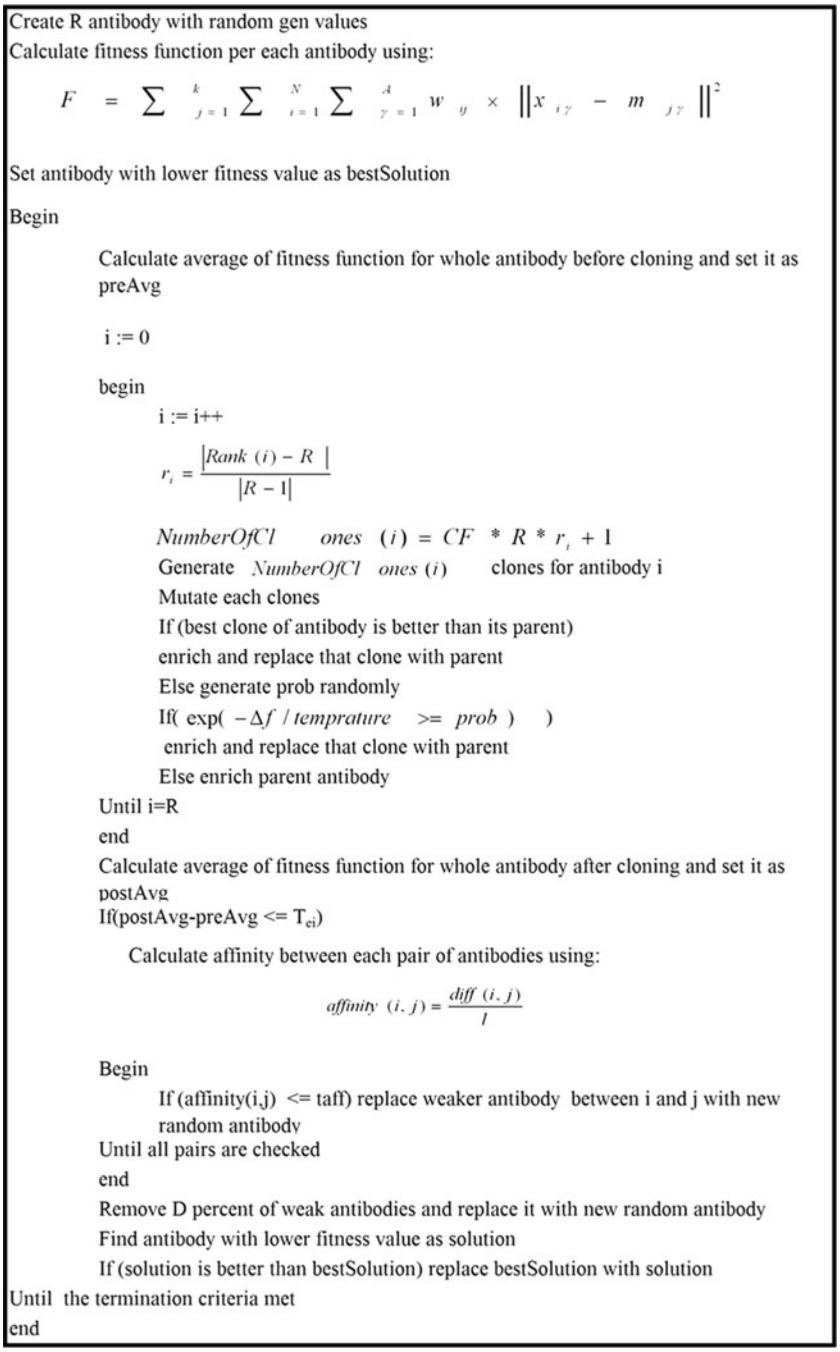
Fig. 2 Nomenclature

Let us assume that we need to assign N objects to K clusters. In order to reach this purpose, R antibodies are used, each of them presenting one of the possible solutions. Figure 1 illustrates proposed antibody string.

Above figure shows a string of solutions including seven objects and three clusters. Each assigned cluster number demonstrates in corresponding cell, e.g., at Fig. 1 first object is assigned to cluster number 3, second is assigned to cluster number 1, and so on.

In this article, summation of Euclidean distances of each object from its associated cluster center is used as

Fig. 3 Pseudo code for IS-SA algorithm



```
Create R antibody with random gen values
Calculate fitness function per each antibody using:
    $F = \sum_{j=1}^{k} \sum_{i=1}^{N} \sum_{\gamma=1}^{A} w_{ij} \times \| x_{i\gamma} - m_{j\gamma} \|^2$
Set antibody with lower fitness value as bestSolution
Begin
        Calculate average of fitness function for whole antibody before cloning and set it as preAvg
        i := 0
        begin
            i := i++
            $r_i = \frac{|Rank(i) - R|}{|R - 1|}$
            $NumberOfCl\ ones(i) = CF * R * r_i + 1$
            Generate $NumberOfCl\ ones(i)$ clones for antibody i
            Mutate each clones
            If (best clone of antibody is better than its parent)
            enrich and replace that clone with parent
            Else generate prob randomly
            If( $\exp(-\Delta f / temprature >= prob)$ )
             enrich and replace that clone with parent
            Else enrich parent antibody
        Until i=R
        end
        Calculate average of fitness function for whole antibody after cloning and set it as postAvg
        If(postAvg-preAvg <= $T_{ei}$)
            Calculate affinity between each pair of antibodies using:
                $affinity(i, j) = \frac{diff(i, j)}{l}$
        Begin
            If (affinity(i,j) <= taff) replace weaker antibody between i and j with new random antibody
        Until all pairs are checked
        end
        Remove D percent of weak antibodies and replace it with new random antibody
        Find antibody with lower fitness value as solution
        If (solution is better than bestSolution) replace bestSolution with solution
Until the termination criteria met
end
```

Table 1 Algorithm parameters and levels

Parameter (symbol)	Index of levels	Levels
Clonal factor (*A*)	*A*(1), *A*(2), *A*(3)	0.05, 0.1, 0.2
Antibody pool size (*B*)	*B*(1), *B*(2), *B*(3)	20, 30, 40
Average error improvement threshold (*C*)	*C*(1), *C*(2), *C*(3)	0.0005, 0.001, 0.002
Affinity threshold (*D*)	*D*(1), *D*(2), *D*(3)	0.05, 0.1, 0.2
Remove rate (*E*)	*E*(1), *E*(2), *E*(3)	0.02, 0.05, 0.1
Enrichment rate (*F*)	*F*(1), *F*(2), *F*(3)	0.3, 0.4, 0.5
Mutation rate (*G*)	*G*(1), *G*(2), *G*(3)	0.25, 0.5, 0.75
Initial temperature (*H*)	*H*(1), *H*(2), *H*(3)	5, 10, 20
Temperature reduction rate (*I*)	*I*(1), *I*(2), *I*(3)	0.999, 0.9995, 0.9999

fitness function. We assume that each object has attributes. Assume that $m_{i\gamma}$ represents the average of the γth attribute of the ith cluster and w_{ij} is an integer variable where if object i is assigned to cluster j is equal to one else is set to zero and $x_{i\gamma}$ is the value of the γth attribute of the ith object.

$$\min F = \sum\nolimits_{j=1}^{k} \sum\nolimits_{i=1}^{N} \sum\nolimits_{\gamma=1}^{A} w_{ij} \times \left\| x_{i\gamma} - m_{j\gamma} \right\|^{2} \tag{1}$$

$$\sum\nolimits_{j=1}^{k} w_{ij} = 1, i = 1, \ldots, N \tag{2}$$

Table 2 Orthogonal array L27

Trial	*A*	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*
1	*A*(1)	*B*(1)	*C*(1)	*D*(1)	*E*(1)	*F*(1)	*G*(1)	*H*(1)	*I*(1)
2	*A*(1)	*B*(1)	*C*(1)	*D*(1)	*E*(2)	*F*(2)	*G*(2)	*H*(2)	*I*(2)
3	*A*(1)	*B*(1)	*C*(1)	*D*(1)	*E*(3)	*F*(3)	*G*(3)	*H*(3)	*I*(3)
4	*A*(1)	*B*(2)	*C*(2)	*D*(2)	*E*(1)	*F*(1)	*G*(1)	*H*(2)	*I*(2)
5	*A*(1)	*B*(2)	*C*(2)	*D*(2)	*E*(2)	*F*(2)	*G*(2)	*H*(3)	*I*(3)
6	*A*(1)	*B*(2)	*C*(2)	*D*(2)	*E*(3)	*F*(3)	*G*(3)	*H*(1)	*I*(1)
7	*A*(1)	*B*(3)	*C*(3)	*D*(3)	*E*(1)	*F*(1)	*G*(1)	*H*(3)	*I*(3)
8	*A*(1)	*B*(3)	*C*(3)	*D*(3)	*E*(2)	*F*(2)	*G*(2)	*H*(1)	*I*(1)
9	*A*(1)	*B*(3)	*C*(3)	*D*(3)	*E*(3)	*F*(3)	*G*(3)	*H*(2)	*I*(2)
10	*A*(2)	*B*(1)	*C*(2)	*D*(3)	*E*(1)	*F*(2)	*G*(3)	*H*(1)	*I*(2)
11	*A*(2)	*B*(1)	*C*(2)	*D*(3)	*E*(2)	*F*(3)	*G*(1)	*H*(2)	*I*(3)
12	*A*(2)	*B*(1)	*C*(2)	*D*(3)	*E*(3)	*F*(1)	*G*(2)	*H*(3)	*I*(1)
13	*A*(2)	*B*(2)	*C*(3)	*D*(1)	*E*(1)	*F*(2)	*G*(3)	*H*(2)	*I*(3)
14	*A*(2)	*B*(2)	*C*(3)	*D*(1)	*E*(2)	*F*(3)	*G*(1)	*H*(3)	*I*(1)
15	*A*(2)	*B*(2)	*C*(3)	*D*(1)	*E*(3)	*F*(1)	*G*(2)	*H*(1)	*I*(2)
16	*A*(2)	*B*(3)	*C*(1)	*D*(2)	*E*(1)	*F*(2)	*G*(3)	*H*(3)	*I*(1)
17	*A*(2)	*B*(3)	*C*(1)	*D*(2)	*E*(2)	*F*(3)	*G*(1)	*H*(1)	*I*(2)
18	*A*(2)	*B*(3)	*C*(1)	*D*(2)	*E*(3)	*F*(1)	*G*(2)	*H*(2)	*I*(3)
19	*A*(3)	*B*(1)	*C*(3)	*D*(2)	*E*(1)	*F*(3)	*G*(2)	*H*(1)	*I*(3)
20	*A*(3)	*B*(1)	*C*(3)	*D*(2)	*E*(2)	*F*(1)	*G*(3)	*H*(2)	*I*(1)
21	*A*(3)	*B*(1)	*C*(3)	*D*(2)	*E*(3)	*F*(2)	*G*(1)	*H*(3)	*I*(2)
22	*A*(3)	*B*(2)	*C*(1)	*D*(3)	*E*(1)	*F*(3)	*G*(2)	*H*(2)	*I*(1)
23	*A*(3)	*B*(2)	*C*(1)	*D*(3)	*E*(2)	*F*(1)	*G*(3)	*H*(3)	*I*(2)
24	*A*(3)	*B*(2)	*C*(1)	*D*(3)	*E*(3)	*F*(2)	*G*(1)	*H*(1)	*I*(3)
25	*A*(3)	*B*(3)	*C*(2)	*D*(1)	*E*(1)	*F*(3)	*G*(2)	*H*(3)	*I*(2)
26	*A*(3)	*B*(3)	*C*(2)	*D*(1)	*E*(2)	*F*(1)	*G*(3)	*H*(1)	*I*(3)
27	*A*(3)	*B*(3)	*C*(2)	*D*(1)	*E*(3)	*F*(2)	*G*(1)	*H*(2)	*I*(1)

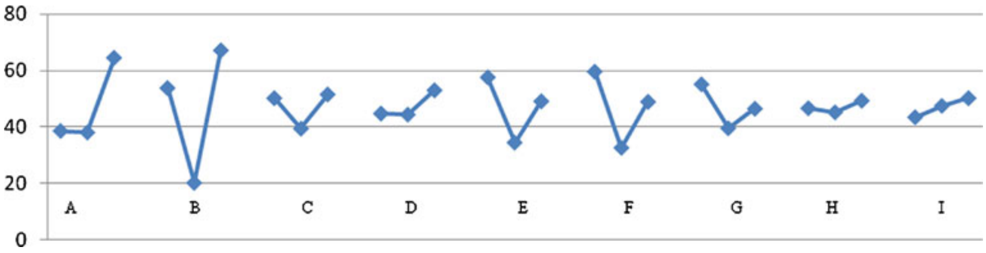


Fig. 4 The average *S*/*N* ratio plot at each level for objective function values

$$\sum_{i=1}^{N} w_{ij} \geq 1, j = 1, \ldots, K \tag{3}$$

$$mj\gamma = \frac{\sum\limits_{i=1}^{N} w_{ij} x_{i\gamma}}{\sum\limits_{i=1}^{N} w_{ij}} \tag{4}$$

The objective function is computed by Eq. 1. Equation 2 implies that each object is assigned to only one cluster. Equation 3 implies that at least one cluster must be assigned to each cluster. Equation 4 calculates average of clusters attributes.

Steps of IS-SA algorithm are as following.

1. Parameter setting. Nomenclature of proposed algorithm is presented at Fig. 3.
2. Random generating of *R* antibodies
3. Fitness function calculating of each antibody and determining best possible solution.

   Meeting termination condition, steps 4 to 9 would be repeated.
4. Calculating average of fitness functions and saving it as *PreAvg*.
5. Clonal selection.

   For each antibody:

   (a) Calculating number of clones which has a direct proportion with related fitness function using subsequent equation:

$$\text{numberOfClones}(i) = \text{CF} \times R \times r_i + 1 \tag{5}$$

Where CF is clonal factor, *R* is number of antibodies and *rank (i)* is the rank of *i*th antibody and:

$$r_i = \frac{|\text{Rank}(i) - R|}{|R - 1|} \tag{6}$$

b. Generating clones which are mutated copies of parent antibodies.

c. Mutating each clone using operator CRHO. CRHO leads to a greater mutation in weaker clones in order to enhance the possibility of improvement. On the other hand, stronger clones suffer from a lesser mutation and subsequently experience lesser variations. Start and end of selected region for mutation is calculated using the following equation:

$$\begin{aligned} \text{start} &= s + (\text{fitnessvalue} - \text{fitAvg}) \\ \text{end} &= e - (\text{fitnessvalue} - \text{fitAvg}) \end{aligned} \tag{7}$$

Where $0 \leq s, e < l$ and l is antibody length.

d. Substituting parent antibody with best clone using Boltzmann acceptance criterion. It means that if the best clone is better than the parent, antibody substitution would be accomplished; else, it is performed with $\text{prob} = \exp(-\Delta f/\text{temperature})$ probability.

Where Δf is the difference between best clone and parent antibody and T is the value of temperature which is reduced using alpha coefficient during each iteration.

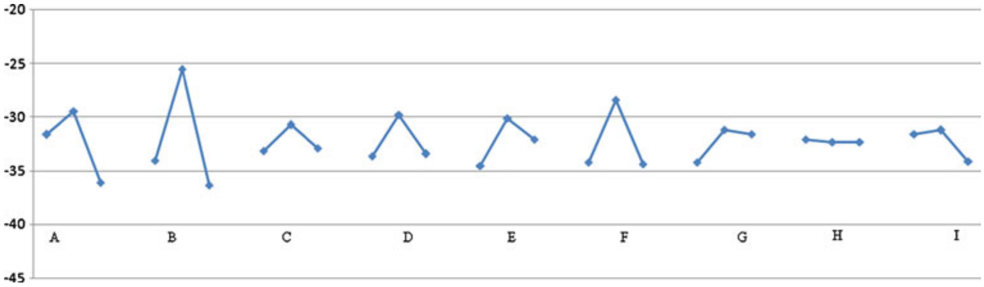


Fig. 5 The average RPD plot at each level for objective function values

Table 3 ANOVA for RPD values

Source	*df*	Seq SS	Adj SS	Adj MS	*F*	*P*
A	2	6,496.2	7,886.82	3,943.41	14.21	0.003
B	2	7,590.5	4,026.19	2,013.09	7.26	0.02
C	2	270.8	48.68	24.34	0.09	0.917
D	2	421.8	984.89	492.45	1.78	0.238
E	2	2,802.8	3,859.96	1,929.98	6.96	0.022
F	2	3,756.2	4,685.94	2,342.97	8.45	0.014
G	2	2,708.4	3,571.39	1,785.7	6.44	0.026
H	2	396.4	671.08	335.54	1.21	0.354
I	2	1,285.2	1,285.24	642.62	2.32	0.169
Residual error	7	1,941.9	1,941.9	277.41		
Total	25	27,670.3				

In the case of determining the best clone for replacement, the proposed enrichment operator with predefined rate implementing on that clone, else such implementation would be performed on parent antibody. Enrichment operator is performed on an antibody and it would be enriched if it has the improvement capability. This operator works as it described below:

Improvement $N\times K$ matrix is constructed in which the *(i,j)* entry indicate the amount of improvement in fitness function corresponding to the assign of object *i* to cluster *j*.

Improvement *(i, j)*=distance of object *i* to the center of a cluster it belongs—distance of object *i* to the center of new cluster if it is transferred to cluster *j*.

Constructing aforementioned matrix, entry of the maximum value indicate how new clusters should be arranged. To do this, the corresponding object of that entry row would be transferred to the corresponding cluster of that entry column.

6. Calculating average of fitness functions and saving it as *PostAvg*.
7. If the value of "*PreAvg-PoAvg*" is equal or less than predefined value of average error improvement threshold (t_{ei}), interactions among antibodies should be considered as follows: Antibodies which are close to each other are recognized and that bigger fitness function are deleted and, then substituted with a new random antibody. If the affinity between two antibodies is less than an affinity *threshold* (t_{aff}), those are recognized as close two antibodies. Hamming distance is considered as affinity criterion which is given by:

$$\mathrm{affinity}(i,j) = \frac{\mathrm{diff}(i,j)}{l} \qquad (8)$$

Where diff*(i,j)*is the number of gens in the *i*th antibody which have different value with corresponding gens of *j*th antibody and l is the antibody length.

Table 4 ANOVA for *S*/*N* ratio

Source	*df*	Seq SS	Adj SS	Adj MS	*F*	*P*
A	2	190.97	244.26	122.128	7.01	0.021
B	2	492.95	271.45	135.727	7.79	0.017
C	2	28.38	58.53	29.265	1.68	0.253
D	2	76.66	121.67	60.836	3.49	0.089
E	2	89.11	99.14	49.572	2.85	0.125
F	2	201.64	251.32	125.662	7.22	0.02
G	2	56.54	89.57	44.785	2.57	0.145
H	2	4.97	16.11	8.056	0.46	0.648
I	2	86.44	86.44	43.22	2.48	0.153
Residual error	7	121.9	121.9	17.414		
Total	25	1,349.56				

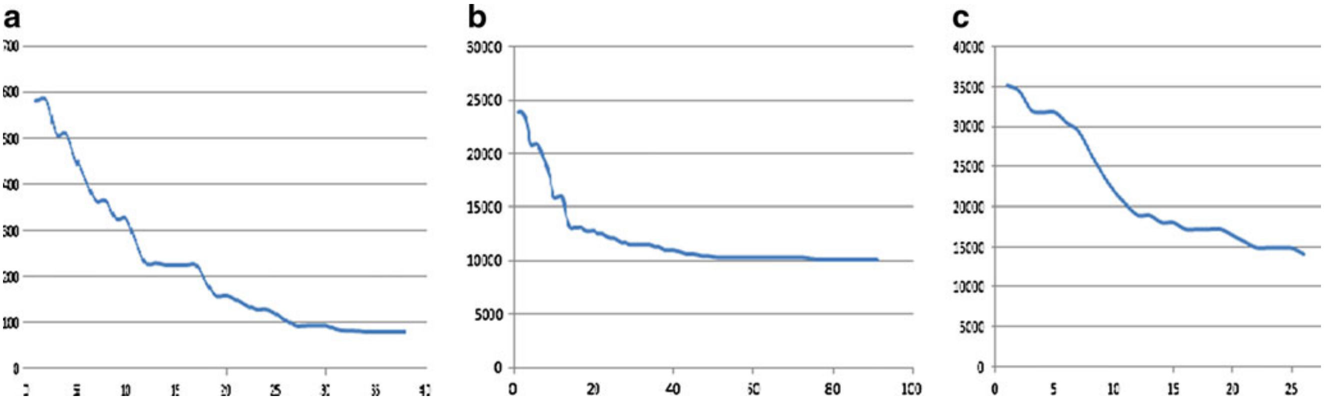


Fig. 6 Convergence of proposed algorithm solution for different data sets: **a**) iris data set, **b**) wine data set, **c**) thyroid diseases data set

8. The best solution should be replaced with the best antibody if that antibody is better.
9. Making diversity in generated solutions, *d* percent of weak antibodies is removed and replaced with new random ones.
10. The algorithm is terminated if a termination criterion (predefined execution time or maximum iteration) met.

Pseudocode of proposed algorithm is presented at Fig. 3 (Figs. 2 and 3).

4 Experimental results

4.1 Taguchi method

In order to tune the parameters of the algorithm, a design of experiment method which was developed by Dr. Taguchi in early 1960s is used [28, 29]. In this method, controllable factors will be posed in inner orthogonal array and noise factors in the outer orthogonal array. So to do, the response values of quality characteristics achieved throughout the experiments will be converted into signal-to-noise ratio (*S*/*N*). The optimal level of parameter can be attained through a further analysis.

Taguchi defined that the optimal operator combination is to minimize variances of quality characteristics resulted from *S*/*N* ratio, which explains the reason why parameter design is also called robust design. In addition, *S*/*N* ratio which is employed for minimizing the variances, the mean of quality characteristics is also utilized for determining the adjustment factors which are used for reason to approach the response variable to the objective point. In general, parameter design processes can be made clear in four steps: the first one is to evaluate the influences of controllable factors on the *S*/*N* ratio and mean of the response. The second is the determination of factors that have significant effect on the *S*/*N* ratio; the level which has the most *S*/*N* ratio will be chosen. The third is the determination of factors that do not significant effect on *S*/*N* ratio and have significant effect on mean of response, and the level whose mean of response is closer to objective point will be chosen. Fourth is the determination of factors which have significant impact neither on *S*/*N* ratio nor on mean of response, are considered as economical factors and the levels that have lowest response will be chosen.

As well, response variable of this paper is relative percentage deviation (RPD) which prefers "the lower is better" principle. Thus, *S*/*N* ratio has the characteristic of "the greater the better". Suitable formula is considered as follows:

$$S/N \text{ ratio} : \eta_j = -10\log\left(\frac{1}{N}\sum\nolimits_{i=1}^{N} y_i^2\right) \qquad (9)$$

Table 5 Result obtained by the five algorithms for ten different runs on dataset 1

Algorithm	F_{best}	F_{average}	F_{worst}	Function evaluation	CPU time (s)
IS-SA	78.945065	79.555264	81.832222	10,439	32.81
HBMO	96.752047	96.95316	97.757625	11,214	35.25
ACO	97.100777	97.171546	97.808466	10,998	33.72
GA	113.986503	125.197025	139.778272	38,128	105.53
TS	97.365977	97.868008	98.569485	20,201	72.86
SA	97.100777	97.134625	97.263845	29,103	95.92
AIS_Lan	98.000126	98.000126	98.000126	16,862	53

Table 6 Result obtained by the five algorithms for ten different runs on dataset 2

Algorithm	F_{best}	F_{average}	F_{worst}	Function evaluation	CPU time (s)
IS-SA	14,879.491839	15,431.426201	15,921.265050	7,065	53.86
HBMO	16,257.284378	16,257.284378	16,257.284378	7,238	55.18
ACO	16,530.533807	16,530.533807	16,530.533807	9,306	68.29
GA	16,530.533807	16,530.533807	16,530.533807	33,551	226.68
TS	16,666.226987	16,785.459275	16,837.535670	22,716	161.45
SA	16,530.533807	16,530.533807	16,530.533807	7,917	57.28

4.2 Parameter tuning

In this section, tuning of those parameters is studied using Taguchi method in order to select the optimum levels of parameter. Factors affecting algorithm performance and their levels are shown in Table 1. In addition, orthogonal array L27 is used for analysis that is shown in Table 2. For tuning the parameters of IS-SA, we use the mean of RPD for Euclidian distance (e). The RPD value is defined as follows:

$$\mathrm{RPD}_{i,j} = \frac{e_{i,j} - \min\limits_{j=1..27} e_{i,j}}{\min\limits_{j=1..27} e_{i,j}} \tag{10}$$

Where i and j denote index of trial and dataset, respectively. Regarding to the results presented in Figs. 4 and 5, levels of 0.1, 30, 0.001, 0.1, 0.05, 0.4, 0.5, 10, and 0.9995 can be considered for A, B, C, D, E, F, G, H, and I, respectively. It can be concluded from Table 3 that determined value of parameters A, B, E, F, and G have the most affect on the algorithm performance. Moreover, similar interpretation can be made about parameters A, B, and F from Table 4.

4.3 Compare IS-SA with other meta-heuristic algorithms

In this section, we present a set of experiments that shows goodness of our algorithm. We have done our experiments on a Pentium IV, 2.1 GHz, 2 GB RAM computer and we have coded with C#.NET. We run all six algorithms on three different datasets. The datasets are all well-known iris, wine, and thyroid diseases datasets taken from Machine Learning Laboratory (http://www.ics.uci.edu/~mlearn/MLRepository.html).

Dataset1: This is the Iris data set, which is perhaps the best-known database to found in the pattern recognition literature. Fisher's paper is a classic in the field and referenced frequently to this day. The data set contains three classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two; the latter are not linearly separable from each other. There are 150 instances with four numeric attributes in iris data set. There is no missing attribute value. The attributes of the iris data set are: sepal length in centimeters, sepal width in centimeters, petal length in centimeters, and petal width in centimeters.

Dataset2: This is the wine data set, which also taken from MCI laboratory. These data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines. There are 178 instances with 13 numeric attributes in wine data set. All attributes are continuous. There is no missing attribute value.

Dataset3: This dataset categories $N=215$ samples of patients suffering from three human thyroid diseases, $K=3$ as: euthyroid, hyperthyroidism, and hypothyroidism patients where 150 individuals are tested euthyroid thyroid, 30 patients are experienced hyperthyroidism thyroid while 35 patients are suffered by hypothyroidism thyroid. The objective function curve for the best solution for each dataset is shown in Fig. 6.

To evaluate the performance of the IS-SA in clustering, we have compared it with several typical stochastic

Table 7 Result obtained by the five algorithms for ten different runs on dataset 3

Algorithm	F_{best}	F_{average}	F_{worst}	Function evaluation	CPU time (s)
IS-SA	10,027.475125	10,049.123921	10,068.682828	23,868	101.91
HBMO	10,099.297958	10,100.13	10,107.646802	23,268	99.35
ACO	10,111.827759	10,112.126903	10,114.819200	25,626	102.15
GA	10,116.294861	10,128.823145	10,148.389608	45,003	153.24
TS	10,249.72917	10,354.315021	10,438.780449	29,191	114.01
SA	10,111.827759	10,114.045256	10,115.934358	28,675	108.22

algorithms including the ACO algorithm [6], the simulated annealing approach [9], the genetic algorithms [8], the Tabu search approach [7], HBMO Algorithm [10], AIS_Lan [5]. It is necessary to be mentioned that AIS_Lan algorithm was merely implemented for iris. The effectiveness of stochastic algorithms is greatly dependent on the generation of initial solutions. Therefore, foe every dataset, algorithms performed ten times individually for their own effectiveness tests, each time with randomly generated initial solutions. The comparison of results for each dataset based on the best solution found in ten distinct runs of each algorithm, the average number of evaluations required and the convergence processing time taken to attain the best solution. The solution quality is also given in terms of the best, average, and worst values of the clustering metric (F_{best}, F_{average}, F_{worst}, respectively) after ten different runs for each of the five algorithms. Tables 5, 6, and 7 show these results.

As it can be observed from the above comparisons, running the IS-SA on the dataset1 and dataset2 results in superior improvement for F_{best}, F_{average}, F_{worst}. Meanwhile its run time is less in comparison with other algorithms. Running the IS-SA on the dataset3 results in trivial improvement. In this case, the run time of the IS-SA is just negligibly (almost 2 s) more than HBMO algorithm.

The result demonstrate that the proposed hybrid algorithm can be considered as a feasible and an efficient heuristic to find optimal or near optimal solutions to clustering problems of allocating N objects to k clusters.

5 Conclusion

The IS-SA algorithm is based on simulated annealing and immune-based methods in which the solution is obtained by employing iterations of cloning, mutation, and enrichment operators as well as considering interactions among antibodies. The enrichment operator is developed to be specifically applied in clustering problem, performing on an antibody to improve its quality. Additionally, Boltzmann criterion is used for accepting the potential solutions. An efficient clustering technique is developed integrating the simulated annealing and evolutionary programming by which the merits of both these approaches are combined. The way by which parameters of proposed algorithm are selected can affect the quality of results. Taguchi method is used for tuning of those parameters in order to select the optimum levels of parameter. In addition, parameters which have the most affect on the algorithm performance are detected. The algorithm implemented and tested on several real datasets. Performance evaluation of the proposed algorithm is studied, comparing with other stochastic algorithms such as ant colony, genetic algorithm, simulated annealing, and Tabu search. Results are very encouraging in terms of the quality of solution found, the average number of function evaluation and the processing time required.

References

1. Han J, Kamber M (2001) Data mining concepts and techniques. Morgan Kaufman, San Francisco
2. Rajendran C, Ziegler H (2004) Ant-colony algorithms for permutation flow-shop scheduling to minimize makespan/total flowtime of jobs. Eur J Oper Res 155:26–38
3. Safari E, Sadjadi SJ, Shahanaghi K (2010) Scheduling flowshops with condition-based maintenance constraint to minimize expected makespan. Int J Adv Manuf Technol 46:757–767
4. C. Kahraman, O. Engin, M.K. Yilmaz (2009), A new artificial immune system algorithm for multiobjective fuzzy flow shop problems, vol. 2, no. 3, pp. 236–247
5. Chen PC, Chen CW, Chiang WL (2009) GA-based modified adaptive fuzzy sliding mode controller for nonlinear systems. Expert Syst Appl 36(3):5872–5879
6. Shelokar PS, Jayaraman VK, Kulkarni BD (2004) An ant colony approach for clustering. Anal Chim Acta 509(1):187–195
7. Sung CS, Jin HW (2000) A tabu-search-based heuristic for clustering. Pattern Recogn 33(3):849–858
8. Mualik U, Bandyopadhyay S (2000) Genetic algorithm-based clustering technique. Pattern Recogn 33(2):1455–1465
9. Selim SZ, Al-Sultan K (1991) A simulated annealing algorithm for the clustering problem. Pattern Recogn 24(10):1003–1008
10. Fathian M, Amiri B, Maroosi A (2007) Application of honey-bee mating optimization algorithm on clustering. Appl Math Comput 190:1502–1513
11. Timmis J, Honec A, Stibord T, Clarka E (2008) Theoretical advances in artificial immune systems. Theoretical Computer Science 403(1):11–32
12. J Timmis, MJ Neal (2000) A resource limited artificial immune system for data analysis. Research and Development in Intelligent Systems XVII, Proceedings of the ES2000, Cambridge, pp. 19–32
13. Tavakkoli-Moghaddam R, Rahimi-Vahed A, Mirzaei A (2007) A hybrid multi-objective immune algorithm for a flow shop scheduling problem with bi-objectives: weighted mean completion time and weighted mean tardiness. Inf Sci 177(22): 5072–5090
14. Kumar A, Prakash A, Shankar R, Tiwari MK (2006) Psycho-clonal algorithm based approach to solve continuous flow shop scheduling problem. Expert Syst Appl 31(3):504–514
15. Tsai J-T, Ho W-H, Liu T-K, Chou J-H (2007) Improved immune algorithm for global numerical optimization and job-shop scheduling problems. Appl Math Comput 94(2):406–424
16. Chandrasekaran M, Asokan P, Kumanan S, Balamurugan T, Nickolas S (2006) Solving job shop scheduling problems using artificial immune system. Int J Adv Manuf Technol 31: 580–593
17. de Castro LN, Timmis J (2002) Artificial immune systems: a new computational intelligence approach. Springer, Berlin
18. Dasgupta D, Forrest S (1996) Novelty detection in time series data using ideas from immunology. Proc ISCA'96, Reno, Nevada, pp. 19–21
19. Forrest S, Hoffmeyr SA (2000) Engineering an immune system. Graft 4(5):5–9
20. Burnet FM (1959) The clonal selection theory of acquired immunity. Cambridge University Press, Cambridge
21. L.N. De Castro, J. Timmis (2002), An artificial immune network for multimodal function optimization, Proceedings of the IEEE Congress

on Evolutionary Computation, vol. 1, IEEE Press, Piscataway, NJ, pp. 674–699

22. J. Kelsey, J. Timmis (2003) Immune inspired somatic contiguous hyper mutation for function optimization. Genetic and Evolutionary Computation Conference. Lecture Notes in Computer Science, vol. 2723. Springer, Berlin, pp. 207–218
23. De Castro LN, Von Zuben FJ (2002) Learning and optimization using the clonal selection principle. IEEE Trans Evol Comput 6(3):239–251
24. Forrest S, Perelson AS, Allen L, Cherukuri R (1994) Self-nonself discrimination in a computer. IEEE Symposium on Research in Security and Privacy. IEEE Computer Society Press, Los Alamos
25. Jerne NK (1974) Towards a network theory of the immune system. Ann Immunol (Inst Pasteur) 125C:373–389
26. De Castro LN, von Zuben FJ (2001) aiNet: an artificial immune network for data analysis. Idea Group Publishing, Hershey, pp 231–259 (Chapter 12)
27. Hiller FS (2003) Handbook of metahueristics. Kluwer, Boston
28. Jeff F, Hamada W, Michael C (2002) Experiments: planning, analysis, and parameter design optimization. Wiley, New York
29. Box G, Draper E, Norman P (2007) Response surfaces, mixtures, and ridge analyses, second edition of empirical model-building and response surfaces. Wiley, New York