International Journal of Management Perspective
Vol.1, No.1
pp. 111-130

Multi-criteria inventory classification problem: An effective artificial immune algorithm

M. Zandieh[1,*]
H. Farmahini Farahani[2]
V. Roshanaei[3]

Abstract

In this paper, we study the problem of multi-criteria ABC inventory classification using an efficient artificial immune algorithm (AIA) to partially alter the traditionally incomprehensive attitude of single objective consideration of inventory control problems. Therefore, we simultaneously endeavor to investigate two different subjects. First, we incorporate various criteria such as annual dollar usage, lead time, criticality, commonality, obsolescence and substitutability into the problem of ABC inventory classification. This method is regarded in lieu of mere consideration of the annual dollar usage criteria in the traditional ABC inventory classification. Second, the proposed AIA delays the algorithm convergence due to its restraining mechanism; meanwhile, it alleviates the problem of premature convergence of existing genetic algorithm to end up with more precise ABC inventory classification. Finally, we draw an analogy between the results obtained from both algorithms applied to a real case study present in the literature. The superiority and effectiveness of our AIA is inferred from all the results obtained in various situations.

Key words: Inventory management, Multi-criteria analysis; ABC classification; Artificial immune algorithm.

[1] Member of research core of World–Class Production Management, Faculty of Management and Accounting, Shahid Beheshti University, G. C., Tehran, Iran.
[2] Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran.
[3] Department of Industrial Engineering, Amirkabir University of Technology, Tehran, Iran
[*] Corresponding author.
E- mail address: m_zandieh@sbu.ac.ir

1. Introduction

Nowadays, in industrial companies ranging from small to large ones, it is a customary policy for them to maintain numerous different items in their inventory sites. There exist wide variety of purposes and objectives to retain these items in inventory and these items encompass for example raw materials, components used in products, items used for activities to support production such as maintenance, cleaning. Newly, companies have encountered the problem of tremendous growth in the number of items that they should be held in their inventory due to their customer pressure needing diverse product and service models. To achieve a noteworthy competitive edge in current and emerging markets, companies need to abide by the orders of their customers and cater to the various taste and demands of their stakeholders within a reasonable amount of time as well as acceptable agility. Rapidly growing demand of customers has brought about many problems for manufacturers which have resulted in dramatic increase in the number of items that carried to the inventory. Operations managers are always seeking an effective type of inventory to assess and assure the timely availability of items when they are needed.

Encountered with the heavy responsibility of handling a numerous number of items, companies need to adopt an effective procedure to simply classify the items in inventory and initiate a drastic inventory control policy to better manage these classes. One of the most popular ways of deciding the importance of inventory items and, thus, an appropriate inventory management method for them is to use ABC classification system which was developed at General Electric during the 1950's. The scheme is based on the Pareto principle of the 18th century economist, Villefredo Pareto, stating that 20% of the people controlled 80% of the wealth. Empirical studies indicate that 5-20% of all items account for 55-65% of total dollar volume, 20-30% of all items account for 20-40% of total dollar volume, and the remaining 50-75% of all items account for only 5-25% of total dollar volume. In the classical ABC classification, items are arranged in descending order of their annual dollar portion values which are the products of annual portion quantities and the average unit prices of the items. The relatively small number of items at the top of the list controlling the majority of the total annual dollar portions constitutes class A and the majority of the items at the bottom of the list controlling a relatively small portion of the total annual dollar usage constitutes class C. Items between the above classes constitute class B. The profiling of these classes is optional and the number of classes may be increased depending on the extent to which a firm contemplates differentiating control efforts. We classify the levels of control into three groups: tight, mediocre and loose. Tight management control of ordering procedures and individual demand forecasts should be applied for class A items i.e. since the items belonging to this group is highly expensive, one should give careful consideration to order them. Class C items should receive a loose control, such as a simple two-bin system, and class B items should have a control effort that reach a compromise between these two extremes. Thus, in a typical firm, concentrating effort should be given to tight control for class A items and a loose one for class C items to obtain sizeable savings. Silver and Peterson (1985) suggested some inventory control policies for the above classes.

Having taken into consideration the curve of the cumulative percentage of total annual dollar usage versus the percentage of items in the ordered list described above, the regions for the classes are easily specified .The curve is an increasing concave one and the regions are simply delineated from each other by the change in slope; the range for the regions is largely reliant to the company, type of industry etc.

The vast application of ABC classification is because of its simplicity, applicability to numerous situations and is also because of the empirically observed advantages on inventory management. However, the procedure has a serious drawback that may inhibit the effectiveness of the procedure in some situations. While making decision regarding inventory procedure, several criteria have to be simultaneously considered to reduce the probability of possible harmful consequences. The sole criterion utilized in the classical ABC classification is the annual dollar usage; considering only one criterion may create problems of significant financial loss. For example, class C items with long lead times or class A items prone to obsolescence may incur financial losses as a result of possible interruption of production and/or huge inventory levels.

2. Literature review

Flores and Whybark (1986) propounded that ABC classification considering multiple criteria, such as lead time, criticality, commonality, obsolescence and substitutability can supply a more exhaustive managerial control. Flores et al. (1992) proposed the consideration of joint criteria matrix only for two criteria. The resulting matrix requires the development of nine different policies, and for more than two criteria it becomes impractical to use the procedure.

Several multiple criteria decision-making (MCDM) tools have also been employed for this specific purpose. Cohen and Ernst (1988) and Ernst and Cohen (1990) have used cluster analysis to group similar items. The analytical hierarchy process (AHP) has been utilized in many multi-criteria inventory classification (MCIC) studies (Flores et al. (1992); Gajpal et al. (1994); Partovi and Burton (1993); Partovi and Hopton (1993)). When AHP is used, the general idea is to derive a single scalar measure of importance of inventory items by subjectively rating the criteria and/or the inventory items (Flores et al. (1992); Guvenir and Erel (1998)). The most important issue associated with AHP-based studies is the subjectivity involved in the analysis. Heuristic approaches based on artificial intelligence, such as artificial neural networks (Partovi and Anandarajan (2002)) and genetic algorithm (Guvenir and Erel (1998)), have also been applied to address the MCIC problem.

One of the very recent attempts by Ramanathan (2006) is to develop a weighted linear optimization model to the problem. The proposed model, hereinafter, is called the R-Model (Peng and Liwei (2007)). The basic concept of R-Model is closely similar to the concept of data envelopment analysis (DEA). R-Model first converts all criteria measures into a scalar score which is a weighted sum of measures under individual criteria. To avoid the subjectivity on the weight assignments, the weights are generated by a DEA-like linear optimization. The classification is then performed by grouping the items based on the scores generated. However, a linear optimization is required for each item. The processing time can be very long when the number of inventory items is large in scale of thousands of items in inventory (Wan Lung (2007)). Wan Lung (2007) proposed simple method for multi criteria ABC analysis that was very similar to R-Model (Ramanathan (2006)). He proposed a simple Model for multiple criteria inventory classification. His model converts all criteria measures of an inventory item to a scalar score. The classification based on the calculated scores using ABC principle is then applied. With proper transformation, one could obtain the scores of inventory items without a linear optimizer. This model has some limitations. One of the limitations of exogenous specification of ranking is the number of criteria and the second limitation is that the model cannot handle discrete measures.

In this paper, an alternative method that uses an artificial immune algorithm (AIA) to learn the weights of criteria is presented. A succinct introduction to the AIA is followed by the description of the alternative multi-criteria inventory classification scheme using an AIA. Finally, we compare the genetic algorithm and the AIA on a sample inventories. The paper concludes with a general discussion of the application of AIA to multi-criteria classification problems.

3. Immune system in general

The natural immune system is a complex adaptive pattern-recognition system that defends the body from foreign pathogens (bacteria or viruses). It is able to categorize all cells (or molecules) within the body as either those belonging to its own kind (self-cell) or those that have a foreign origin (non self-cell) (Dasgupta (2002)). It has dramatic and complex mechanisms that recombine the gene to cope with the invading antigens, produce the antibodies and exclude the antigens (Mori et al. (1993)). The infection process involves invasion of a pathogen and its proliferation within the organism. Pathogens are associated with specific proteins (antigens). The immune system contains cells that are capable of recognizing antigens and killing pathogens. These cells, further referred to as immune cells (antibodies), are randomly distributed throughout the immune system. In the relatively evolutionarily advanced animals, the immune system is capable of enhanced response to re-infection by an earlier encountered pathogen (adaptive immunity).

Each individual immune cell involved in adaptive immunity is capable of recognizing only one type of antigen. Therefore, there is a huge diversity of immune cells in the organism waiting for many possible antigens.

In the case of infection only a small proportion of immune cells would react, i.e., those are pre-programmed for this particular antigen. This interaction triggers fast multiplication of these particular cells (clonal proliferation). The number of immune cells capable of recognizing the specific antigens and killing the specific pathogen increases by many orders. Thus, the organism's immune system becomes tuned to fight not just random pathogens but specifically the one that actually invaded (Gutnikov and Melnikov (2003)). The clonal selection and affinity maturation principles are used to explain how the immune system reacts to pathogens and how it improves its capability of recognizing and eliminating pathogens (Ada and Nossal (1987)). Clonal selection states that when a pathogen invades the organism, a number of immune cells that recognize these pathogens will proliferate; some of them will become effecter cells, while others will be maintained as memory cells. The effecter cells secrete antibodies in large numbers, and the memory cells have long life spans so as to act faster and more effectively in future exposures to the same or a similar pathogen. During cellular reproduction, the cells suffer somatic mutations at high rates, together with a selective force; the cells with higher affinity to the invading pathogen differentiate into memory cells. This whole process of somatic mutation plus selection is known as affinity maturation (De Castro and Timmis (2002)).

The natural immune system is a very complex system with several mechanisms to defend against pathogenic organisms. However, the natural immune system is also a source of inspiration for solving optimization problems. From the information processing perspective, immune system is a remarkable adaptive system and can provide several important aspects in the field of computation (Frank et al. (1996); Dasgupta and Attoh-Okine (1997)). When incorporated with evolutionary algorithms, immune system can improve the search ability during the evolutionary process (Jiao and Wang (2000)).

4. An artificial immune algorithm to multi-criteria inventory classification

Here we propose an alternative method to learn the weight vector along with the cut-off values for multi-criteria inventory classification. The method proposed here, called AIAMIC (artificial immune algorithm for multi-criteria inventory classification), uses an AIA to learn the weights of criteria along with AB and BC cut-off points from pre-classified items.

In AIAs, antigens refer to the objective function that needs to be optimized. The antibodies refer to the candidate solutions to a problem. Usually, initial antibodies are randomly generated on a feasible space.

The exploration of new antibodies is generally implemented by generating a pool of candidate solutions in an iterative manner until a pre-determined number of generations are obtained. An affinity calculation between antibodies is also embedded within the algorithm to suppress similar antibodies. Through immune evolutionary algorithm (IEA) computation, an antibody that most fits the antigen is considered as the solution to multi-criteria inventory classification.

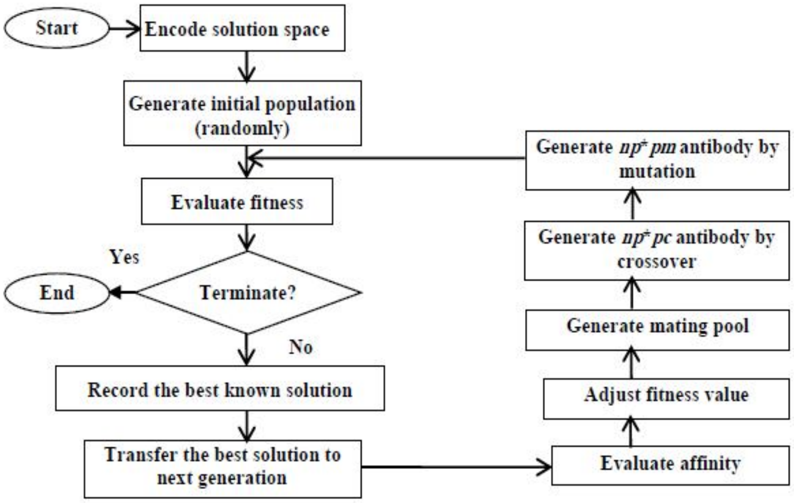

Start
Encode solution space
Generate initial population (randomly)
Evaluate fitness
Yes
End
Terminate?
No
Record the best known solution
Transfer the best solution to next generation
Evaluate affinity
Adjust fitness value
Generate mating pool
Generate np*pc antibody by crossover
Generate np*pm antibody by mutation

Fig.1. Flowchart of proposed AIA approach to multi-criteria inventory classification.

Fig. 1 shows a flowchart of an AIA to multi-criteria inventory classification. The proposed IEA steps are as follows:

Step 1. Initialization:

(a) Parameters setting: Set the number of initial population (np), number of generation (ng), probability of crossover (pc), probability of mutation (pm), affinity threshold (at), and affinity adjustment (aa).

(b) Randomly generation of an initial population of np antibodies.

Step 2. Objective function evaluation: Evaluate the fitness for each antibody.

Step 3. Termination test: Terminate the algorithm if the stopping criterion is met; else return to Step 2.

Step 4. Mating pool generation:

(a) The best antibody selection: Record the best known antibody based on fitness in mating pool (*accelerating mechanism*).

(b) Affinity evaluation: Evaluate the similarity between each antibody with the best known antibody obtained so far.

(c) Similar antibodies suppression: If the affinity value for each antibody is more than a prescribed threshold at, then reduce the probability assigned to those multiplying by aa and normalize the probability (*restraining mechanism*).

(d) Mating pool expansion: Select with replacement $np-1$ antibodies from the full population (including the best antibody). The antibodies are selected according to their fitness, with those antibodies having a higher fitness value being selected more often.

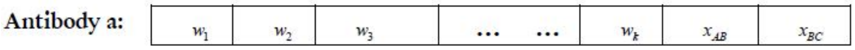
Step 5. Crossover operation: Select $np \times pc$ pairs of parents from mating pool, and perform crossover on the parents at random.

Step 6. Replacement: While retaining the best antibody from the previous generation, replace the remaining $np-1$ antibodies with the current population of clones (or offspring) from Step 3.

Step 7. Mutation operation: Select $np \times pm$ antibodies from mating pool, and mutate the individual bits. Return to Step 2.

4.1. Encoding scheme

An antibody with the length of $k+2$(with k sub-segment) constitutes and encodes the weight vector together with two cut-off points. The values of sub-segments fall within the interval of real values 0 and 1. The aggregation of all elements in the weight vector always amounts to 1($\sum_{j=1}^{k} w_j = 1$). Additionally, the AB cut-off value (x_{AB}) always exceeds the BC cut-off value (x_{BC}). Hence, if the classification is based on k criteria, an antibody **a** is a vector defined as



Antibody a:	w_1	w_2	w_3	… …	w_k	x_{AB}	x_{BC}

Here, w_j represents the weight of the *j*-th criterion, $\sum_{j=1}^{k} w_j = 1$, and $x_{BC} < x_{AB}$. This type of representation enables the antibody to encode the relative weight of a criterion as an absolute value. That is, the weight of the corresponding criterion is represented by the value of a sub-segment independent of the other value of sub-segments.

Given an antibody **a**, classification of an inventory item i is done by computing its weighted sum, $ws(\mathbf{a}, j)$ as follows:

$$ws(\mathbf{a}, i) = \sum_{j=1}^{k} w_j \frac{i_j - \min j}{\max j - \min j}$$

Here i_j is the value of the item i for the criterion j, $\max j$ and $\min j$ are maximum and minimum values of criterion j among all inventory items, respectively. The classification of an inventory item i according to antibodies **a** is

$$Cl(\mathbf{a}, i) = \begin{cases} A & ; if \;\; x_{AB} \le ws(\mathbf{a}, i) \\ B & ; if \;\; x_{BC} \le ws(\mathbf{a}, i) < x_{AB} \\ C & ; Otherwise \end{cases}$$

Taking into consideration this encoding scheme, AIAMIC applies the standard immune operators (reproduction, crossover, and mutation) to the antibodies in the population. AIAMIC applies fitness proportionate roulette wheel selection strategy in crossover. AIAMIC also utilizes the elitist selection strategy in reproduction, i.e. the best antibody is always copied to the next generation. Below, the evaluation of the fitness of an antibody, the crossover and the mutation operations, selection scheme and similar antibodies suppression employed by AIAMIC is explicated.

4.2. Fitness function computation

The fitness evaluation of an antibody is directly influenced by its ability to precisely classify the training set. Therefore, any misclassified item should introduce (is subject to) a penalty. However, due to the linear ordering among the classes, we have to distinguish the error made by classifying a class A item as a class B item than as a class C item. In our implementation the fitness of an antibody **a** was computed as

$$fitness(\mathbf{a}) = \frac{1}{t} \sum_{i=1}^{t} p_i$$

$$p_i = \begin{cases} 1 & Cl(i, \mathbf{a}) = class(i) \\ 0.4 & |Cl(i, \mathbf{a}) - class(i)| = 1 \\ 0 & otherwise \end{cases}$$

where t is the size of the training set, and $class(i)$ is the classification given to the *i*-th training instance by the decision maker. Note that this fitness function prefers an antibody making a single mistake with a difference of 2 to an antibody making two mistakes with difference 1.

4.3. Crossover operator

The most important operation in an AIA is the crossover operation. Randomly selected pairs of antibodies go under crossover operation with a fixed probability, pc. Although the initial population is setup in a way that all antibodies represent legal coding (the value of each sub-segment is between 0 and 1, the sum of all weight values is 1, and the AB cut-off is less than the BC cut-off), standard crossover operations are bound to result in illegal coding.

Here we use a new form of uniform crossover operation for structures that are vectors of continuous values, called *continuous uniform crossover,* which guarantees the legality of the offspring. Given two antibodies $\mathbf{x} = (x_1, x_2, ..., x_{k+2})$ and $\mathbf{y} = (y_1, y_2, ..., y_{k+2})$, the offsprings are defined as $\mathbf{x}' = (x_1', x_2', ..., x_{k+2}')$ and $\mathbf{y}' = (y_1', y_2', ..., y_{k+2}')$, where

$$x_i' = \lambda x_i + (1 - \lambda) y_i$$

$$y_i' = (1 - \lambda) x_i + \lambda y_i$$

Here λ is constant through a single crossover operation. This crossover preserves the sum of any subset of sub-segments. In the case of inventory classification, if the sub-segments 1 through k encode the criteria weights, then $\sum_{i=1}^{k} x_i = 1$. After the crossover operation,

$$\sum_{i=1}^{k} x_i' = \lambda \sum_{i=1}^{k} x_i + (1 - \lambda) \sum_{i=1}^{k} y_i = \lambda + (1 - \lambda) = 1$$

This is true for both offsprings $\sum_{i=1}^{k} x'_i = 1$ and $\sum_{i=1}^{k} y'_i = 1$. Furthermore, this crossover preserves the greater-than relation between sub-segments, as well. That is, if $x_{AB} > x_{BC}$, then $x'_{AB} > x'_{BC}$.Therefore, continuous uniform crossover preserves the legality of antibodies for the multi-criteria ABC classification.

The choice of the λ is an important issue. If $\lambda = 0$, the offsprings are the same as the parents. If $\lambda \geq 0$ then the values of sub-segments remain to be between 0 and 1, however, the values of sub-segments get closer to each other through generations. If $\lambda = 0.5$, then both offsprings are the same, and the values of sub-segments are equal to the average values of the respective sub-segments in the parents. On the other hand, if $\lambda < 0$, the values of sub-segments diverge from the respective values in the parents. However, if $\lambda < 0$, then sub-segment values may be outside of the limits; that is each value of sub-segment may get a negative value of a value greater than 1, although their sum is still 1. In that case, a normalization of the antibody is needed. For $\lambda < 0$, after the crossover we check if any sub-segment value is less than 0 (if any sub-segment value is greater than l, then there exists at least one sub-segment value less than 0). In that case we first subtract the minimum sub-segment value from all sub-segments value, then set

$$x_i = \frac{x_i}{\sum_{j=1}^{k} x_j}$$

where k is the number of criteria. Again, if $\lambda < 0$, then x_{AB} may be smaller than x_{BC}; in that case we swap the values of x_{AB} and x_{BC}. In our implementation we chose λ randomly from [-0.5, 0.5] for each crossover operation.

4.4. Mutation operator

The mutation operation in our implementation sets the value of a sub-segment to either 0 or 1, with equal probability. If an antibody is modified by the mutation operation, it has to be normalized as in the case of the uniform crossover operation with $\lambda < 0$.

4.5. Selection scheme and similar antibodies suppression

Selection is the process by which bad members are culled out of the current population. All members are ranked for goodness based on a fitness function. After ranking, a selection function uses each antibody's goodness value to determine which antibodies will survive to the next generation. The method that is used to select members for survival is important to achieve the best solution, and greatly influences the diversity of solution and rate of solution convergence.

The candidate solutions are then compared with the best-known solutions obtained so far. This allows an affinity value, which expresses the similarity between candidate solutions with a known best solution, to be computed using entropy theory. The details of the entropy theory are outlined in a later section. Candidate solutions with a higher fitness value are given a higher probability to be selected for the production of the next generation of candidate solutions. The assignment of probability values is done using the roulette wheel method (Goldberg (1953)). To assist the search, an AIA uses both an *accelerating mechanism* and a *restraining mechanism*. Basically, the *accelerating mechanism* works as follows; a candidate solution with the best fitness is recorded, which will be used to seed the mating pool. This mechanism is to ensure that the mating pool contains a large proportion of candidate solutions with good properties. On the other hand, the *restraining mechanism* performs the following tasks. If a candidate antibody has an affinity value higher than a prescribed threshold value, then these two antibodies are called similar ones. To maintain diversity, the probability assigned to each solution is multiplied by a factor which is less than one. This will reduce the probability of being selected. Such a mechanism is built into an AIA to prevent a good candidate antibody from becoming overly dominant. Without the restraining mechanism, a good candidate antibody may be able to multiply and dominate the mating pool prematurely. As a result, it may restrict the search space for candidate antibodies (Zandieh et al. (2006)).

4.6. Affinity computation

Entropy theory is employed here to estimate the probability of the recurrence of patterns from an information source. Abramson (1963) defined the information entropy, $H(x)$, of a discrete random variable

$X = \{x_1, x_2, x_3, \ldots\ldots, x_n\}$ with probability mass function $P(X = x_i) = p_i$, $i = 1, 2, \ldots, n$ as $H(x) = -\sum_{i=1}^{n} p_i . \log p_i$.

Using information entropy, the similarity of antibody i (or sequence i) in relation to a reference antibody (or sequence) can be expressed as follows:

$$aff(i) = \frac{1}{1 + \frac{1}{k}\sum_{j=1}^{k} h_{ij}}$$

Where, $aff(i)$ is the measure of similarity, k is the sequence size, and $h_{ij} = -p_{ij} \log p_{ij}$.

If $\left| x_j - x_{j\,ref} \right| \leq 0.01$ then $p_{ij} = 1$; thus $h_{ij} = -\log 1 = 0$. On the other hand, if $\left| x_j - x_{j\,ref} \right| > 0.01$ then $p_{i\,j} = 0.5$; this implies that $h_{ij} = -0.5 \log 0.5 = 0.151$.

It can be inferred from above that two identical sets of numbers would yield an affinity value of 1.00. Conversely, two completely different sets of numbers would give an affinity value of $\frac{1}{1+0.151} \approx 0.87$; such an observation applies to sequences of numbers of any length.

5. Empirical comparison

In order to exemplify the application of the AIAMIC, we here evaluate its behavior on a sample inventory classification task which was used by Guvenir and Erel (1998). We also provide the application of the genetic algorithm on the same tasks for the comparison. Although the sample inventories used here are relatively small, the size of the inventory has no effect on the applicability of these methods.

Similar to the Guvenir and Erel (1998) study, we used a human decision maker who is actually responsible for the inventory. The classification accuracy of both AIAMIC and genetic algorithm were measured in terms of their similarity to the classification of the decision maker.

5.1. University stationery inventory

Here, we reproduce an example from the literature to demonstrate the executive application of our algorithm on the problem at hand (Guvenir and Erel (1998)). This example involves the stationery items held in the stockroom of the Purchasing Department of a medium size University. The Department has got the responsibility of procuring, receiving, and keeping inventory as well as deciding on the timing, procurement order sizes and on-hand inventory levels. It is a highly centralized department in which most of the major decisions are made by a manager; hereinafter the manager is called the decision maker. The decision maker has the full authority to develop procurement and inventory stocking policies for the items held in the department. The inventory consists of 145 stationery items.

The decision maker considers four criteria to classify the items, as follows:

C_1: Annual dollar usage.

C_2: Number of requests for the item in a year.

C_3: Lead time.

C_4: Substitutability. (0: substitutable, 1: sometimes substitutable; 2: cannot be substituted).

To constitute the training set, the decision maker is requested to name ten items from each of the three classes. This set of 30 items forms our training set.

5.2. Artificial immune algorithm parameters tuning

It is known that the different levels of the parameters directly affect the quality of the solutions obtained by an AIA. A number of different AIAs can be obtained with the different combinations of the parameters. We have applied parameter tuning for the crossover probability (pc), mutation probability (pm), affinity threshold (at), affinity adjustment (aa), and the number of generations (ng). We took into account the following ranges:

- Crossover probability (pc): three levels (0.55, 0.70, 0.85).
- Mutation probability (pm): three levels (0.1, 0.05, 0.025).
- Affinity threshold (at): three levels (0.92, 0.95, 0.98).
- Affinity adjustment (aa): three levels (0.50, 0.70, 0.90).

- Number of generations (ng): two level (150, 300).
- Number of initial population (np): one level (100).

AIA was experimented based on full factorial experiment design. The full factorial experiment design for aforesaid five factors requires $3^4 * 2^1 = 162$ experiments, so 162 different level combinations of control factors were considered. For each trial, eight replications of AIA were performed, then obtained results of total fitness are recorded. The fitness of total number of generated solutions is averaged in each level and its value is plotted against each control factor in Fig. 2.

As indicated in Fig. 2 better fitness of the algorithm is achieved when the parameters are set as follows: $aa:0.7$; $pc:0.7$ and $at:0.95$. Meanwhile, other parameters are set as below. The $ng:300$ and $pc:0.05$. The analysis of variance (ANOVA), given in Table 1, is carried out for statistical significance test of factors. Hence, ANOVA is carried out again after pooling of factors the pm and ng in the error terms. The results, given in Table 2, indicate that factors aa, pc and at give significant impact on the fitness of algorithm. Based on Table 2, it is obvious that aa, pc and at factors with 40.3%, 22.7% and 9% respectively have got the largest impact on the average fitness of the algorithm.

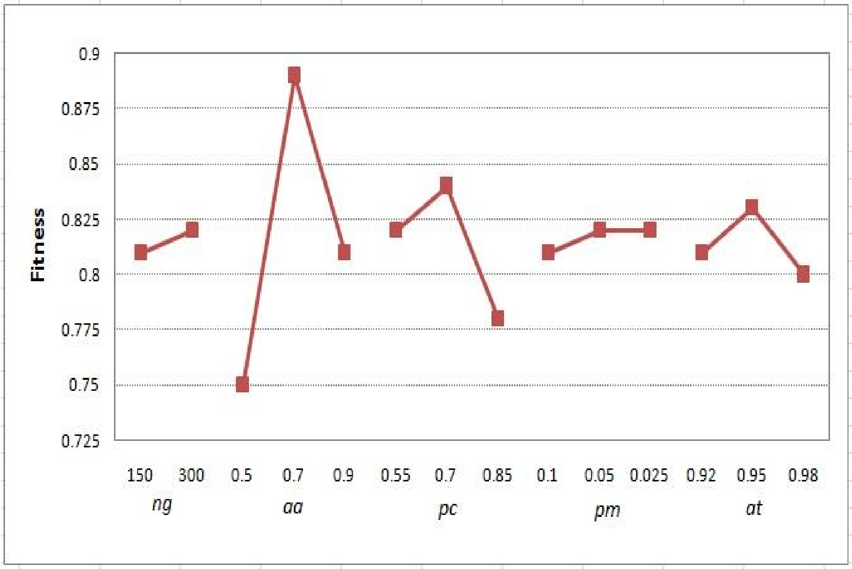
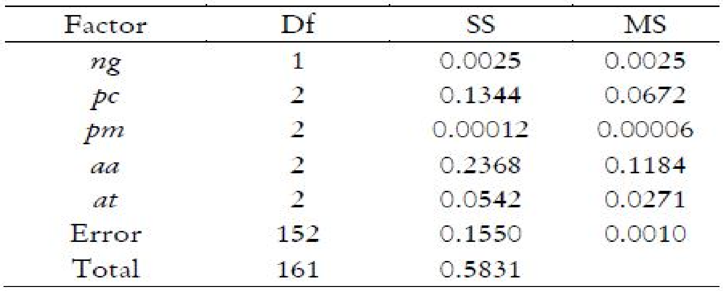


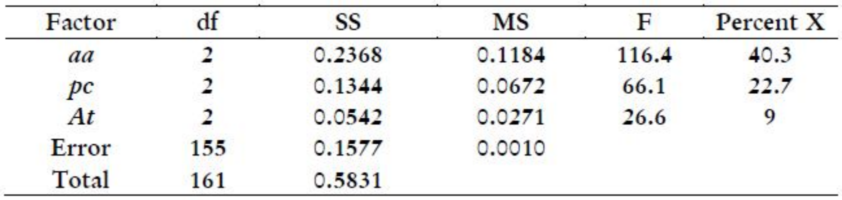
Fig.2. The average fitness plot at each level of factors.

Table 1. ANOVA for fitness average



Factor	Df	SS	MS
ng	1	0.0025	0.0025
pc	2	0.1344	0.0672
pm	2	0.00012	0.00006
aa	2	0.2368	0.1184
at	2	0.0542	0.0271
Error	152	0.1550	0.0010
Total	161	0.5831	

Table 2. ANOVA for fitness average



Factor	df	SS	MS	F	Percent X
aa	2	0.2368	0.1184	116.4	40.3
pc	2	0.1344	0.0672	66.1	22.7
At	2	0.0542	0.0271	26.6	9
Error	155	0.1577	0.0010		
Total	161	0.5831			

5.3. Experimental results

Both algorithms AIAMIC and GAMIC were coded in MATLAB. Then, the parameters of algorithm AIAMIC was set to $pc = 0.7,\ pm = 0.05,\ ng = 300,\ aa = 0.7$ and $at = 0.95$ and GAMIC was set in conformity with the work of Guvenir and Erel (1998). Finally, we applied these algorithms on the training set. The best and average fitness values in each generation of AIAMIC and GAMIC were recorded. Fig. 3 and Fig. 4, depict the convergence speed of AIAMIC and GAMIC respectively. Based on Fig. 3 and Fig. 4, premature convergence of GAMIC in comparison with AIAMIC is quite obvious (approximately 40*th* and 206*th* generations for GAMIC and AIAMIC respectively).

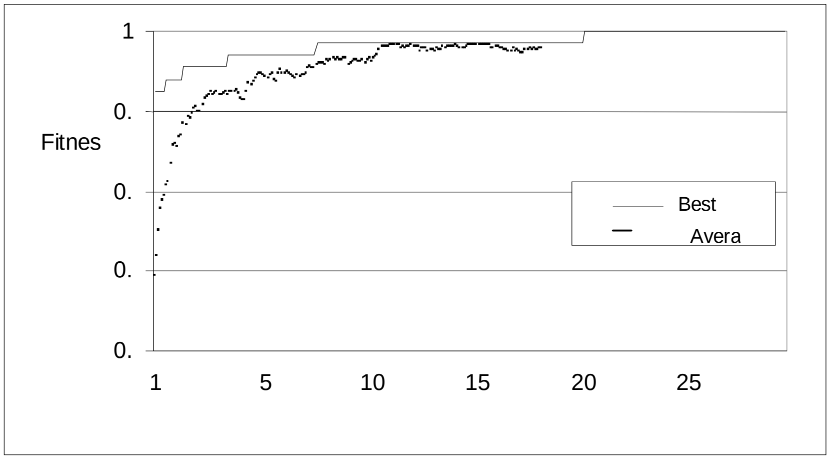


Fig.3. Best and average fitness values through generations of AIAMIC.

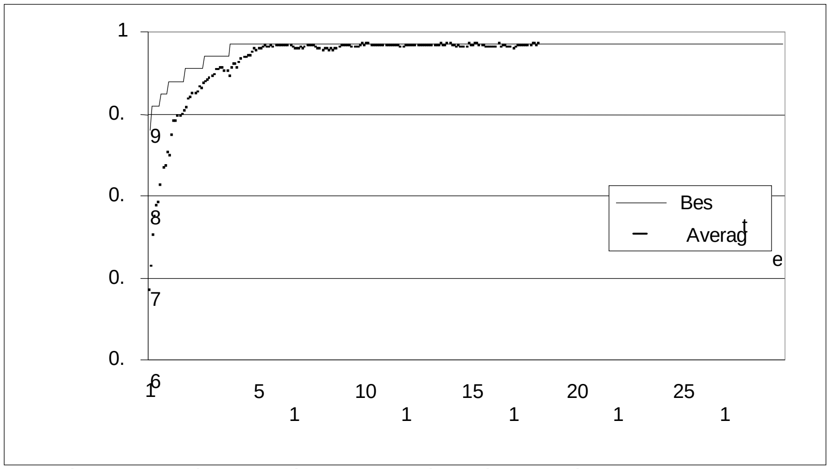
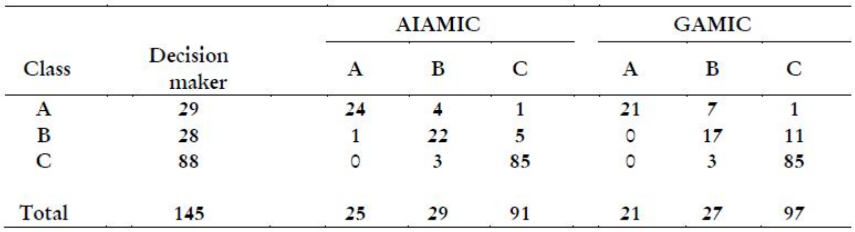


Fig.4. Best and average fitness values through generations of GAMIC.

We classified the remaining 115 items using the weights and cut-off values learned by the AIAMIC. The decision maker also was requested to classify these test items. The decision maker's classification did not agree with AIAMIC on 10 items. The results of the classifications by AIAMIC versus the decision maker are given in Table 3.

Also the results of the classifications by GAMIC vis-à-vis the decision maker are given in Table 3. The decision maker did not agree with the classification of 22 items. A comparison of fitness values made by ten runs of the AIAMIC and ten runs of GAMIC is given in Fig. 5. As a result we can conclude that the proposed AIAMIC outperformed GAMIC in our test set.

Table 3.Comparison of classifications made by the AIAMIC and GAMIC with respect to decision makers opinion on the stationary inventory[1]



Class	Decision maker	AIAMIC			GAMIC		
		A	B	C	A	B	C
A	29	24	4	1	21	7	1
B	28	1	22	5	0	17	11
C	88	0	3	85	0	3	85
Total	145	25	29	91	21	27	97

[1]. Entry in *i*-th row and *j*-th column represents the number of classifications made by decision maker as class *i* and by the corresponding algorithm as class *j*.

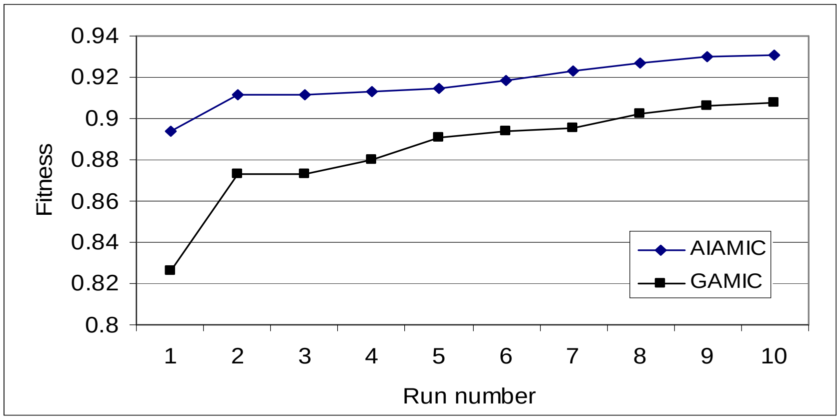


Fig.5. Best fitness values through ten runs of AIAMIC and GAMIC.

6. Conclusion and future works

We applied a new approach using artificial immune algorithms to a multi-criteria inventory classification problem. This approach is based on learning a weight vector of absolute weights of each criterion along with a set of cut-off points. In order to apply an AIA to the weight learning problem, we used a new crossover operator that guarantees the generation of offsprings that are valid representations of weight vectors.

The approach, implemented in a program called AIAMIC, is applicable to any multi-criteria classification problem with any number of classes, provided that it is possible to reduce the problem to learning a weight vector along with the cut-off points between classes. We compared our approach with the GAMIC. The classifications made by AIAMIC were much closer to the classification made by the decision maker than the one obtained by the GAMIC. We believe that the approach presented here is transferrable to other classification problems as well.

References

Abramson, N., (1963). *Information Theory and Coding*. New York: McGrawHill,.

Ada, G.L., Nossal, G.J.V., (1987). The clonal selection theory. *Scientific American, 257*, 50–57.

Cohen, M.A., Ernst, R., (1988). Multi-item classification and generic inventory stock control policies. *Production and Inventory Management Journal, 29,*3, 6–8.

Dasgupta, D., Attoh-Okine, N., (1997). Immunity-based systems: a survey. *Proceedings of the IEEE International Conference on Computational Cybernetic Simulation, 1,* 369–374.

Dasgupta, D., (2002). Special issue on artificial immune system, *IEEE Transactions on Evolutionary Computation, 6*, 225–256.

De Castro, L.N., Timmis, J., (2002). An artificial immune network for multimodal function optimization. *Evolutionary Computation, CEC'02, in: Proceedings of the 2002 Congress,* 699–704.

Ernst, R., Cohen, M.A., (1990). Operations related groups (ORGs): a clustering procedure for production / inventory systems. *Journal of Operations Management, 9*, 4, 574–98.

Flores, B.E., Olson, D.L., Dorai, V.K., (1992). Management of multi-criteria inventory classification. Mathematical and Computer Modeling, 16 (12), 71-82.

Flores, B.E., Whybark, D.C., (1986). Multiple criteria ABC analysis. *International Journal of Operations and Production Management, 6*, 3, 38-46.

Frank, S.A., Rose, M.R., Lauder, G.V., (1996). *The Design of Natural and Artificial Adaptive Systems*. New York: Academic Press.

Gajpal, P.P., Ganesh, L.S., Rajendran, C., (1994). Criticality analysis of spare parts using the analytic hierarchy process. *International Journal of Production Economics, 35*,1–3, 293–7.

Goldberg, D.E., (1953). *Genetic Algorithms in Search Optimization and Machine Learning*. Reading, MA: Addison-Wesley.

Gutnikov, S., Melnikov, Y., (2003). A simple non-linear model of immune response. *Chaos, Solitons & Fractals*, *16*, 125–132.

Guvenir, H.A., Erel, E., (1998). Multi-criteria inventory classification using a genetic algorithm. *European Journal of Operational Research, 105*, 29–37.

Jiao, L., Wang, L., (2000). Novel genetic algorithm based on immunity. *IEEE Transaction System Manufacturing Cybernetic, Part A 30*,5, 552–561.

Mori, K., Tsukiyama, M., Fukuda, T., (1993). Immune algorithm with searching diversity and its application to resource allocation problem. *Transactions of IEE Japan 113C*, 872–878.

Partovi, F.Y., Anandarajan, M., (2002). Classifying inventory using an artificial neural network approach. *Computers and Industrial Engineering, 41*, 389–404.

Partovi, F.Y., Burton, J., (1993). Using the analytic hierarchy process for ABC analysis. *International Journal of Production and Operations Management, 1*,9, 29–44.

Partovi, F.Y, Hopton, W.E., (1993). The analytic hierarchy process as applied to two types of inventory problems. *Production and Inventory Management Journal, 35*,1, 13–9.

Peng, Z., Liwei, F, (2007). A note on multi-criteria ABC inventory classification using weighted linear optimization. *European Journal of Operational Research, 182*, 1488–1491.

Ramanathan, R., (2006). ABC inventory classification with multiple-criteria using weight linear optimization. *Computers and Operations Research, 33*, 695–700.

Silver E.A. Peterson R., (1985). *Decision Systems for Inventory Management and Production Planning*. New York: Wiley.

Wan Lung, Ng., (2007). A simple classifier for multiple criteria ABC analysis. *European Journal of Operational Research, 177*, 344–353.

Zandieh, M., Fatemi Ghomi, S.M.T., Moattar Husseini, S.M. (2006). An immune algorithm approach to hybrid flow shops scheduling with sequence-dependent setup times. *Applied Mathematics and Computation, 180*, 111–127.