Multicriteria Inventory Classification Using A Genetic Algorithm

H. Altay Guvenir [a,1], and Erdal Erel [b]
[a] *Department of Computer Engineering and Information Science*
[b] *Department of Business Administration,*
Bilkent University, 06533 Ankara, Turkey

Abstract

One of the application areas of genetic algorithms is parameter optimization. This paper addresses the problem of optimizing a set of parameters that represent the weights of criteria, where the sum of all weights is 1. A chromosome represents the values of the weights, possibly along with some cut-off points. A new crossover operation, called *continuous uniform crossover*, is proposed, such that it produces valid chromosomes given that the parent chromosomes are valid. The new crossover technique is applied to the problem of multicriteria inventory classification. The results are compared with the classical inventory classification technique using Analytical Hierarchy Process.

Keywords: Computers, Genetic algorithms, Analytical hierarchy process, Inventory, Multi criteria analysis

1. Introduction

It is not uncommon to observe companies of even moderate sizes to carry thousands of different items in inventory. These items are kept for various purposes and include raw materials, components used in products, items used for activities to support production such as maintenance, cleaning, etc. Recently, the number of items carried in inventory increased drastically with the increase of customer pressure demanding different models of products. In order to obtain a competitive advantage in the market, companies have to respond to satisfy the various demands of customers; this results in an increased number of different items carried in inventory in smaller quantities.

Faced with the heavy burden of dealing with a large number of items, companies must classify the items in inventory and develop effective inventory control policies for these classes. The classical ABC classification, developed at General Electric during the 1950's, is the most popular scheme to classify the items in inventory. 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

[1] Corresponding Author. Email: guvenir@cs.bilkent.edu.tr

volume. In the classical ABC classification, items are ordered in the descending order of their annual dollar usage values which are the products of annual usage 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 usage 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 designation of these classes is arbitrary and the number of classes may be increased depending on the extent to which a firm wants to differentiate control efforts. Tight management control of ordering procedures and individual demand forecasts should be made for class A items. 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 lies between these two extremes. Thus, in a typical firm, concentrating effort on tight control for class A items and a loose one for class C items result in substantial savings. Silver and Peterson [14] suggested some inventory control policies for the above classes.

The regions for the classes are easily specified by examining the curve of the cumulative percentage of total annual dollar usage versus the percentage of items in the ordered list described above. The curve is an increasing concave one and the regions are typically distinguished from each other by the change in slope; the range for the regions depends on the company, type of industry, etc.

The wide popularity of the procedure is due to its simplicity, applicability to numerous situations and the empirically observed benefits on inventory management. However, the procedure has a serious drawback that may inhibit the effectiveness of the procedure in some situations. The criterion utilized in the classical ABC classification is the annual dollar usage; using one criterion only 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.

It has been suggested by Flores and Whybark [4] that ABC classification considering multiple criteria, such as lead time, criticality, commonality, obsolescence and substitutability can provide a more comprehensive managerial control. To tackle the difficulties of using only one criterion, Flores et al. Whybark [5] have proposed the use of joint criteria matrix 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.

Analytical Hierarchy Process (AHP) developed by Saaty [13] has been successfully applied to multicriteria inventory classification by Flores et al. [5]. They have used AHP to reduce multiple criteria to a univariate and consistent measure. However, Flores et al. have taken average unit cost and annual dollar usage as two different criteria among others. The problem with this approach is that the annual dollar usage and the unit price of items are usually measured in different units. For example, unit prices of some items are given in $/kgs, while some in $/meters. On the other hand, for the applicability of this approach, the unit of a criterion must not change from item to item. For that reason, we combined these two criteria in one criterion as the total annual dollar usage, of which the measuring unit is dollars.

In this paper we propose an alternative method that uses a *genetic algorithm* to learn the weights of criteria. In the sequel, we first discuss the outline of AHP and its application in multicriteria inventory classification. A brief introduction to the genetic algorithms is followed by the description of the alternative multicriteria inventory classification scheme using a genetic algorithm. Finally, we compare the AHP and the genetic algorithm approaches on two sample inventories. The paper concludes with a general discussion of the application of genetic algorithms to multicriteria classification problems.

2. AHP in Multicriteria Inventory Classification

AHP is proposed by Saaty [13] to aid decision makers in situations involving a finite set of alternatives and multiple objectives. The procedure has been successfully applied to numerous areas, such as marketing [3, 12], forecasting foreign exchange rates [16], among others; Zahedi [17] discusses the application areas of the procedure. The procedure consists of the following steps:

1. In the first step the pairwise comparison matrix A is constructed. This is the most crucial step of the procedure in the sense that the input of the decision maker about criteria preferences is assessed; the quality of the output depends directly on this step. The decision maker is asked to assign a value in a scale 1 through 9 for each pair of criteria. These pairwise comparisons are combined in a square matrix called pairwise comparison matrix. Given a comparison matrix A, the entry A_{ij} represents the relative strength of criterion C_i when compared to C_j. Thus, A_{ij} is expected to be w_i/w_j, where w_i and w_j are the relative strengths of the criteria C_i and C_j, respectively. A value of 1 for A_{ij} represents that criteria C_i and C_j to be of equal importance, whereas a value of 9 represents criterion C_i to be absolutely more important than criterion C_j. Note that $A_{ij} = 1/A_{ji}$ with the diagonal values equal to 1.

2. In this step, the consistency of the pairwise comparison matrix A is checked, first. The matrix is consistent if $A_{ik} = A_{ij} \cdot A_{jk}$ for all $i, j, k \leq n$, where n is the size of the matrix (number of criteria). Saaty [13] has shown that if the diagonal of a matrix A consists of ones ($A_{ii} = 1$) for all i, and if A is consistent, then small variations of the A_{ij} keep the largest eigenvalue λ_{max} close to n, and the remaining eigenvalues close to zero. Therefore, if A is a matrix of pairwise comparison values, in order to find a weight vector, a vector $\mathbf{w}$ that satisfies $A\mathbf{w} = \lambda_{max}\mathbf{w}$ is to be found.

   Inconsistency of a pairwise comparison matrix is due to the inconsistent comparisons of the decision maker. In that case, the decision maker is asked to modify the matrix repeatedly until the matrix is consistent.

3. The next step is the computation of a vector of priorities, $\mathbf{w} = < w_1, w_2, ..., w_k >$. The eigenvector of the comparison matrix with the largest eigenvalue provides the priority ordering of the criteria, that is their weights. The values of this vector are between 0 and 1, and their sum is 1. Saaty [13] has presented some approximate methods to compute the eigenvector of a matrix, since the computation of the exact eigenvector is complex and time-consuming. Among the approximate methods, the

following one results the best approximation: Divide each entry in column i by the sum of the entries in column i, and estimate the vector of largest eigenvalues by taking the average of the entries in row i. For each item in inventory, the criteria values are organized in a way that the higher values increase the probability of an item being in class A.

4. As the criteria have different units of measure, the measures have to be converted to a common 0-1 scale and the weighted score of each item is computed as

$$ws(\mathbf{w}, i) = \sum_{j=1}^{k} w_j \frac{i_j - min_j}{max_j - min_j}.$$

   where k is the number of criteria, i_j is the value of item i for criterion j, max_j and min_j are maximum and minimum values of criterion j among all items, respectively.

5. Finally the items in the inventory are sorted in the decreasing order of their weighted sum. The classification of items in this sorted list is determined by specifying the cutoff points; for example, the first 20% of the items are classified as class A, the next 20% as class B, and the remaining as class C.

The procedure is based on a few assumptions of which should be checked prior to applying the technique to the problem on hand. The first assumption is about the measuring units of criteria: The units of the items must be identical for the criterion considered. The assumption imposes that the units of items for a specific criterion should not differ from item to item. The difficulty caused by different units can be overcome by combining the several criteria to create a new one. For example, if the average unit cost and annual usage are taken as two separate criteria, the units for the first criterion would stay the same for the items as dollars per item, but the units for the second criterion might vary from item to item, such as meters per year, kgs per year, etc. If these two criteria is combined by multiplying them to create the criterion of annual dollar usage, the units of this criterion would be identical for all the items as dollars per year.

The second assumption is about the value function of the decision maker; it is assumed that the decision maker has an additive value function. In other words, the mutually preferentially independence of the objectives should be checked with the decision maker. If the value function has a nonadditive form, then combining some criteria to create a new one, analogous to the above example, may satisfy the assumption.

Finally, the comparisons of the decision maker are assumed to be consistent; otherwise, several attempts may be required to obtain consistent pairwise comparisons. It might be quite difficult to obtain a consistent matrix.

The method is a useful and powerful tool to construct mathematical models of problems that involve multiple objectives and qualitative factors such as subjective judgments. Hence, it has been quite successful to satisfy the needs of decision makers in various environments as demonstrated by the wealth of reports published about the method in the literature. Zahedi gives a comprehensive survey of the method and its applications [17]. On the other hand, there are some drawbacks that may cause serious difficulties in practice. One of the important drawbacks of the method is that a significant amount of subjectivity is involved in pairwise comparisons. Consequently, the resulting classification may be far away from satisfying the objective of the company. Different decision makers may assess different comparison matrices due to various reasons. Another drawback of the method is that it is quite difficult to compare two criteria and assign a numerical value. It may be difficult to assign a numerical value to the figure of speech used to emphasize the comparisons, such as "strongly" or "demonstrably" more important. Furthermore, the range of the scale values being between 1 and 9 causes difficulties. Adopting a different scale may lead to a different classification. The scale and its implications should be explained fully to the decision maker prior to any analysis. Different decision makers may prefer different scales.

In the following section we discuss some properties of genetic algorithms and present an alternative inventory classification scheme with multiple criteria using a genetic algorithm.

3. Genetic Algorithms

Genetic algorithms are general-purpose search algorithms that use principles inspired by natural population genetics to evolve solutions to problems [11]. Genetic algorithms have been applied to a wide range of optimization and learning problems, including routing and scheduling [2, 15], engineering design optimization [1, 6], curve fitting [8] and machine learning [9]. The reader is referred to [7] for a thorough coverage of genetic algorithm applications in various areas.

The basic idea in genetic algorithms is to maintain a population of knowledge structures (also called *chromosomes*) that represent candidate solutions to the current problem. A chromosome is a sequence of *genes*. The population evolves over time through competition and controlled variation. The initial population can be initialized using whatever knowledge is available about possible solutions. Each member of the population is evaluated and assigned a measure of its *fitness* as a solution. After evaluating each structure in the population, a new population of structures is formed in two steps. First, structures in the current population are selected for replication based on their relative fitness. High-performing structures might be chosen several times for *reproduction*, while poorly performing structures might not be chosen at all. Next, the selected structures are altered using idealized *genetic operators* to form a new set of structures for evaluation. The primary genetic search operator is the *crossover* operator, which combines the features of two *parent* structures to form two *offsprings* similar to the parents. The role of the crossover over operation is to form new fit chromosomes from fit parents. There are many possible forms of crossover: the simplest swaps corresponding segments of a string, list or vector representation of the parents. In generating new structures for testing, the crossover operator usually draws only on the information present in the structures of the current knowledge base. If specific information is missing, due to storage limitations or loss incurred during the selection process of a previous generation, then crossover cannot produce new structures that contain it. A *mutation* operator which alters one or more components of a selected structure, provides the way to introduce new information into the knowledge base. A wide range of mutation operators have been proposed, ranging from completely random alterations to more heuristically motivated local search operators. In most cases, mutation serves as a secondary search operator that ensures the reachability of all points in the search space.

The resulting offsprings are then evaluated and inserted back into the population. This process continues until either, an absolute fittest structure is detected in the population or a predetermined stopping criterion (e.g., maximum number of generations or maximum number of fitness evaluations) is reached.

Genetic algorithms can be used in parameter optimization problems by encoding a set of parameter values in a structure of the population. In that case, each structure represents a possible set of parameters of the system being optimized. Then, a fittest structure that represents the optimum setting of parameters is searched via a genetic algorithm.

4. Application of GA to Multicriteria Inventory Classification

Here we propose an alternative method to learn the weight vector along with the cut-off values for multicriteria inventory classification. The method proposed here, called GAMIC (for Genetic Algorithm for Multicriteria Inventory Classification), uses a genetic algorithm to learn the weights of criteria along with AB and BC cut-off points from preclassified items. Once the criteria weights are obtained, the weighted scores of the items in the inventory are computed similarly to the approach with AHP. Then the items with scores greater than AB cut-off value are classified as A, those with values between AB and BC as class B, and the remaining items are classified as class C.

4.1. Encoding

A chromosome encodes the weight vector along with two cut-off points. The values of the genes are real values between 0 and 1. The total value of the elements of the weight vector is always 1. Also the AB cut-off value (x_{AB}) is always greater than the BC cut-off value (x_{BC}). Therefore, if the classification is based on k criteria, a chromosome $\mathbf{c}$ is a vector defined as

$$\mathbf{c} = < w_1, w_2, \ldots w_k, x_{AB}, x_{BC} > .$$

Here, w_j represents the weight of jth criterion, $\sum_{j=1}^{k} w_j = 1$, and $x_{BC} < x_{AB}$. In this representation the relative weight of a criterion can be encoded as an absolute value in a chromosome. That is, an allele (value of a gene) represents the weight of the corresponding criterion, independent of the other alleles.

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

$$ws(\mathbf{c}, i) = \sum_{j=1}^{k} w_j \frac{i_j - min_j}{max_j - min_j}.$$

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. The classification of an inventory item i according to chromosome $\mathbf{c}$ is

$$classification(\mathbf{c}, i) = \begin{cases} A & \text{if } x_{AB} \leq ws(\mathbf{c}, i) \\ B & \text{if } x_{BC} \leq ws(\mathbf{c}, i) < x_{AB} \\ C & \text{otherwise} \end{cases}$$

Given this encoding scheme, GAMIC applies the standard genetic operators (reproduction, crossover, and mutation) to the chromosomes in the population [7]. GAMIC applies fitness proportionate roulette wheel selection in reproduction. GAMIC also uses the elitist approach, that is the best chromosome is always copied to the next generation. The evaluation of the fitness of a chromosome, the crossover, and the mutation operations employed by GAMIC are described below.

4.2. Fitness function

The fitness of a chromosome reflects its ability to classify the training set correctly. Therefore, any misclassified item should introduce 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 a chromosome $\mathbf{c}$ was computed as

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

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

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

4.3. Crossover

The most important operation in a genetic algorithm is the *crossover* operation. Randomly selected pairs of chromosomes go under crossover operation with a fixed probability, p_c. Although the initial population is setup in a way that all chromosomes represent legal codings (each allele 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 codings. For example, consider the following two chromosomes:

$$\begin{aligned} \mathbf{x} = & < 0.1\ 0.4\ 0.3 \mid 0,2 > \\ \mathbf{y} = & < 0.1\ 0.3\ 0.1 \mid 0,5 > \end{aligned}$$

A classic 1-point crossover operation would yield the following offsprings:

$$\begin{aligned} \mathbf{x}' = & < 0.1\ 0.4\ 0.3\ 0,5 > \\ \mathbf{y}' = & < 0.1\ 0.3\ 0.1\ 0,2 > \end{aligned}$$

Since $\sum_{j=1}^{4} x'_j = 1.3 > 1$ and $\sum_{j=1}^{4} y'_j = 0.7 < 1$, both $\mathbf{x}'$ and $\mathbf{y}'$ represent illegal weight settings.

Here we propose 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 offsprings.

Continuous Uniform Crossover: Given two chromosomes $\mathbf{x} = < x_1, x_2, \ldots, x_n >$, and $\mathbf{y} = < y_1, y_2, \ldots, y_n >$, the offsprings are defined as $\mathbf{x}' = < x'_1, x'_2, \ldots, x'_n >$, and $\mathbf{y}' = < y'_1, y'_2, \ldots, y'_n >$, where

$$\begin{aligned} x'_i &= s x_i + (1-s) y_i \\ y'_i &= (1-s) x_i + s y_i \end{aligned}$$

Here s, called *stride*, is constant through a single crossover operations. This crossover preserves the sum of any subset of genes. In the case of inventory classification, if the genes 1 through m encode the criteria weights, then $\sum_{i=1}^{m} x_i = 1$. After the crossover operation,

$$\sum_{i=1}^{m} x'_i = s \sum_{i=1}^{m} x_i + (1-s) \sum_{i=1}^{m} y_i = s + (1-s) = 1.$$

This is true for both offsprings; $\sum_{i=1}^{m} x'_i = 1$, and $\sum_{i=1}^{m} y'_i = 1$, Further, this crossover preserves the greater-than relation between genes, as well. That is, if $x_{AB} > x_{BC}$, then $x'_{AB} > x'_{BC}$. Therefore, continuous uniform crossover preserves the legality of chromosomes for the multicriteria ABC classification.

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

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

where k is the number of criteria. Again, if $s < 0$ 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 s randomly from [-0.5, 0.5] for each crossover operation.

4.4. Mutation

Mutation operation in our implementation sets the value of a gene to either 0 or 1, with equal probability. If a chromosome is modified by the mutation operation, it has to be normalized as in the case of uniform crossover operation with $s < 0$.

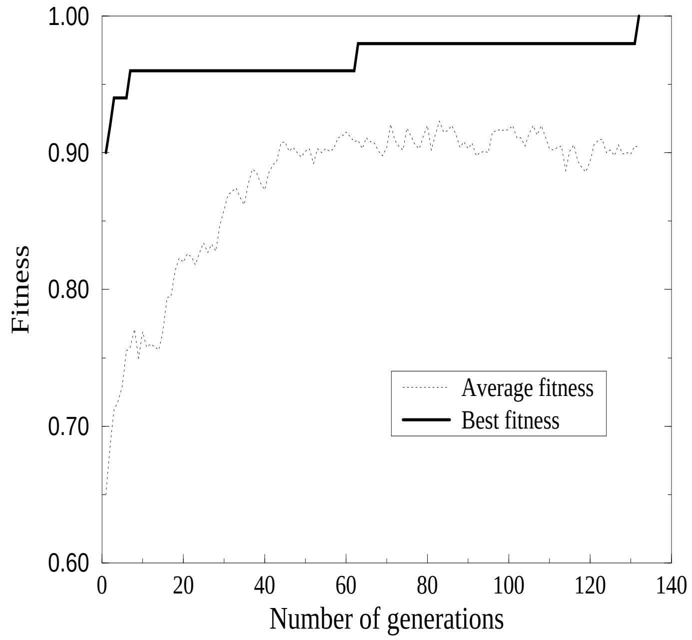

Figure 1: Best and average fitness values through generations in the stationary inventory.

5. Empirical Comparisons

In order to exemplify the application of the GAMIC algorithm, here we present its behavior on two small sample inventory classification tasks. We also provide the application of the AHP approach on the same tasks for a comparison. Although the sample inventories used here are relatively small, the size of the inventory has no effect on the applicability of these methods.

In these experiments, we used a human decision maker who is actually responsible for the inventory. The classification accuracy of both GAMIC and AHP algorithms were measured in terms of their similarity to the classification of the decision maker.

5.1. University Stationary Inventory

This example involves the stationary items held in the stockroom of the Purchasing Department of a medium size University. The Department is responsible 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; hereafter we will call the manager as 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 stationary items.

Table 1: Comparison of classifications made by the GAMIC and AHP with respect to decision maker, on the stationary inventory. Entry in the ith row and jth column represents the number of classifications made by decision maker as class i and by the corresponding algorithm as class j.

Decision Maker		GAMIC			AHP		
		A	B	C	A	B	C
A	26	23	3	0	12	8	6
B	27	1	21	5	11	12	4
C	92	0	1	91	6	9	77
Total	145	24	25	96	29	29	87

The decision maker classified the items according to four criteria, as follows:

C_1: Annual dollar usage
C_2: Number of requests for the item in a year
C_3: Lead time
C_4: Replacability (0: replaceable, 1: sometimes replaceable; 2: cannot be replaced)

We first asked the decision maker to name ten items from each of the three classes. In order to minimize subjectivity, we also requested the decision maker to make his selection from the characteristic and distinguishing items. This set of 30 items formed our training set for the GA. The genetic algorithm was run on a population of 100 chromosomes, with $p_c = 0.7$ and $p_m = 0.001$. Our genetic algorithm converged on the training set in the 132nd generation after 9285 function evaluations. During this process GAMIC has performed 4580 crossover operations and 75 mutations. The best and average fitness values in each generation are shown in Figure 1. The weights learned by GAMIC are as follows: $w_1 = 0.268, w_2 = 0.359, w_3 = 0.034, w_4 = 0.339$ and the cut-off points are $x_{AB} = 0.347, x_{BC} = 0.228$.

We classified the remaining 115 items using the weights and cut-off values learned by the GA. The decision maker also was asked to classify these test items. The decision maker's classification did not agree with GAMIC on 10 items. The results of the classifications by GAMIC versus the decision maker are given in Table 1.

In order to compare with the AHP technique, the decision maker was asked to pairwise compare the four criteria. In the third attempt the decision maker was able to obtain a consistent matrix. The weights obtained by the AHP technique are as follows: $w_1 = 0.558, w_2 = 0.233, w_3 = 0.156, w_4 = 0.052$ and the cut-off points are $x_{AB} = 0.157, x_{BC} = 0.113$. All 145 items were then classified according to the weights learned by the AHP technique. The results of the classifications by AHP versus the decision maker are given in Table 1. The decision maker did not agree with the classification of 44 items. As a result we conclude that the genetic algorithm method proposed here performed much better than the AHP technique in our test set.

5.2. Explosive Inventory

This example involves the items held in the site warehouse of a company that undertakes rock excavation tasks using explosives [10]. The number of criteria used in the classification of the items in that inventory is larger relative to the previous example inventory:

C_1:	Unit price
C_2:	Number of requests for the item in a year
C_3:	Lead time
C_4:	Scarcity
C_5:	Durability
C_6:	Substitutability
C_7:	Repearability
C_8:	Order size requirement
C_9:	Stockability
C_{10}:	Commonality

The criteria C_4 through C_{10} take integer values between 0 and 5. The criteria values are assigned to items in such a way that higher values increase the likelihood of an item to be placed in class A.

The characteristics of the 115 items held in the inventory vary significantly. For example, a highly expensive explosive with a short shelf-life (low durability), and a particular size nail with high substitutability, stockability, commonality, and durability are in the same inventory.

The decision maker in this example is the manager of the warehouse, who is the sole responsible person in maintaining the inventory.

In a similar manner, we first asked the decision maker to name five items from each of the three classes. In order to minimize subjectivity, we also requested the decision maker to make his selection of the above 15 items in such a way that he is absolutely certain about the classes each item belongs to. This set of 15 items formed our training set for the GA. The genetic algorithm was run using the same set of parameters; that is, population of 100 chromosomes, with $p_c = 0.7$ and $p_m = 0.001$. Our genetic algorithm converged on the training set in the 340th generation after 24003 function evaluations. During this process GAMIC has performed 11888 crossover operations and 427 mutations. The best and average fitness values in each generation are shown in Figure 2. The weights learned by GAMIC are as follows: $w_1 = 0.151, w_2 = 0.009, w_3 = 0.276, w_4 = 0.137, w_5 = 0.072, w_6 = 0.168, w_7 = 0.000, w_8 = 0.013, w_9 = 0.035, w_{10} = 0.138$ and the cut-off points are $x_{AB} = 0.476, x_{BC} = 0.325$.

We classified the remaining 100 items using the weights and cut-off values learned by the GA. The decision maker also was asked to classify these test items. The decision maker's classification did not agree with GAMIC on 5 items. The results of the classifications by GAMIC versus the decision maker are given in Table 2.

In order to compare with the AHP technique, the decision maker was asked to pairwise compare the four criteria. In the fourth attempt the decision maker was able to obtain a consistent matrix. The weights obtained by the AHP technique are as follows: $w_1 =$

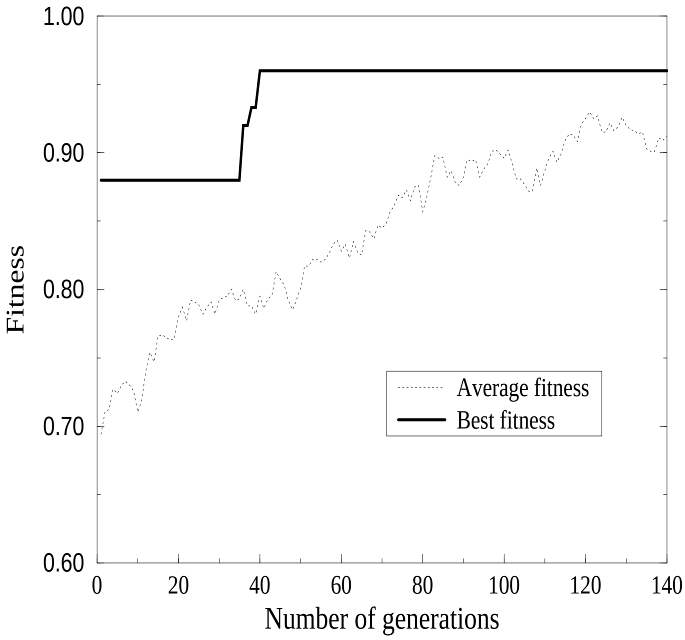

Figure 2: Best and average fitness values through generations in the explosives inventory.

$0.358, w_2 = 0.137, w_3 = 0.246, w_4 = 0.066, w_5 = 0.041, w_6 = 0.018, w_7 = 0.008, w_8 = 0.057, w_9 = 0.043, w_{10} = 0.025$ and the cut-off points are $x_{AB} = 0.33, x_{BC} = 0.15$. All 115 items were then classified according to the weights learned by the AHP technique. The results of the classifications by AHP versus the decision maker are given in Table 2. The decision maker did not agree with the classification of 15 items. As a result we conclude that the genetic algorithm method proposed here performed much better than the AHP technique in our test set.

The GAMIC program is coded in the LISP language. The program took 70 seconds and 151 seconds of CPU time for stationary and explosives inventory examples, respectively, on a SunSparc 20/61 workstation.

6. Conclusion

We presented a new approach using genetic algorithms to multicriteria classification. This approach is based on learning a weight vector of absolute weights of each criteria along with a set of cut-off points. In order apply a genetic algorithm to the weight learning problem, we proposed a new crossover operator that guarantees the generation of offspring that are valid representations of weight vectors.

The approach, implemented in a program called GAMIC, is applicable to any multicriteria classification problem with any number of classes, provided that it is possible

Table 2: Comparison of classifications made by the GAMIC and AHP with respect to decision maker, on the explosives inventory. Entry in the ith row and jth column represents the number of classifications made by decision maker as class i and by the corresponding algorithm as class j.

Decision Maker		GAMIC			AHP		
		A	B	C	A	B	C
A	26	26	0	0	22	4	0
B	24	1	23	0	1	18	5
C	65	0	4	61	0	5	60
Total	115	27	27	61	23	27	65

to reduce the problem to learning a weight vector along with the cut-off points between classes. We have compared our approach with classical AHP technique on two sample inventory classification tasks. The classification made by GAMIC were much closer to the classification made by the decision-maker than the one obtained by the AHP technique. We have seen that decision makers are more comfortable in classifying inventory items than comparing two criteria on a scale of 1 to 9. Therefore, the decision-makers' decisions about individual classifications of items are more reliable. The main advantage of the approach employed in GAMIC over the AHP technique is its more reliable input. We believe that the approach presented here is applicable to other classification problems, as well.

References

[1] Balakrishnan, P.V., and Jacob, V.S., "Triangulation in Decision Support Systems: Algorithms for Product Design", *Decision Support Systems* 14 (1995) 313-327.

[2] Bean, J.C., "Genetics and Random Keys for Sequencing and Optimization", *ORSA Journal on Computing* 6/2 (1994) 154-160. University of Michigan, Ann Arbor, 1993.

[3] Davies, M.A., "Using the AHP in marketing decision-making", *Journal of Marketing Management* 10/1-3 (1994) 57-74.

[4] Flores, B.E., and Whybark, D.C., "Multiple Criteria ABC Analysis", *International Journal of Operations and Production Management* 6/3 (1986) 38-46.

[5] Flores, B.E., Olson, D.L., and Dorai, V.K., "Management of Multicriteria Inventory Classification", *Mathematical and Computer Modeling* 16/12 (1992) 71-82.

[6] Goldberg, D.E., and Samtani, M.P., "Engineering optimization via genetic algorithm", *Proceedings of the Ninth Conference on Electronic Computation*, 1986, 471-482.

[7] Goldberg, D.E., *Genetic Algorithms in Search, Optimization & Machine Learning*, Reading, Massachusetts, Addison-Wesley, 1989.

[8] Gulsen, M., Smith, A.E., and Tate, D.M., "A Genetic Algorithm Approach to Curve Fitting", *International Journal of Production Research* 33/7 (1995) 1911-1923.

[9] Guvenir, H.A., and Sirin, I., "A Genetic Algorithm for Classification by Feature Partitioning", *Proceedings of the Fifth International Conference on Genetic Algorithms*, S. Forest (Ed.), 1993, 543-548.

[10] Guvenir, N., "Application of AHP to Multicriteria Inventory Classification", MBA Thesis, Graduate School of Business Administration, Bilkent University, Ankara, Turkey, 1993.

[11] Holland, J.H., *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, Mass., 1975.

[12] Lu, M.H., Madu, C.N., and Kuei, C.H., "Integrating QFD, AHP and benchmarking in strategic marketing", *The Journal of Business and Industrial Marketing* 9/1 (1994) 41-50.

[13] Saaty, T.L., *The Analytic Hierarchy Process*, New York, McGraw-Hill, 1980.

[14] Silver, E.A., and Peterson, R., *Decision Systems for Inventory Management and Production Planning*, New York, John Wiley and Sons, 1985.

[15] Uckun, S., Bagchi, S., Kawamura, K., and Miyabe, Y., "Managing Genetic Search in Job Shop Scheduling", *IEEE Expert* (October 1995) 15-24.

[16] Ulengin, B., and Ulengin, F., "Forecasting foreign exchange rates: A comparative evaluation of AHP", *Omega* 22/5 (1994) 505-519.

[17] Zahedi, F., "The analytical hierarchy process - a survey of the method and its applications", *Interfaces* 16/4 (1996) 96-108.