Journal of Intelligent Manufacturing (2000) **11,** 435–451

Clustering and selection of multiple criteria alternatives using unsupervised and supervised neural networks

BEHNAM MALAKOOTI and VISHNU RAMAN

Department of Electrical Engineering and Computer Science, Case Western Reserve University, Cleveland, OH 44106, USA

There are decision-making problems that involve grouping and selecting a set of alternatives. Traditional decision-making approaches treat different sets of alternatives with the same method of analysis and selection. In this paper, we propose clustering alternatives into different sets so that different methods of analysis, selection, and implementation for each set can be applied. We consider multiple criteria decision-making alternatives where the decision-maker is faced with several conflicting and non-commensurate objectives (or criteria). For example, consider buying a set of computers for a company that vary in terms of their functions, prices, and computing powers.

In this paper, we develop theories and procedures for clustering and selecting discrete multiple criteria alternatives. The sets of alternatives clustered are mutually exclusive and are based on (1) similar features among alternatives, and (2) preferential structure of the decision-maker. The decision-making process can be broken down into three steps: (1) generating alternatives; (2) grouping or clustering alternatives based on similarity of their features; and (3) choosing one or more alternatives from each cluster of alternatives.

We utilize unsupervised learning clustering artificial neural networks (ANN) with variable weights for clustering of alternatives, and we use feedforward ANN for the selection of the best alternatives for each cluster of alternatives. The decision-maker is interactively involved by comparing and contrasting alternatives within each group so that the best alternative can be selected from each group. For the learning mechanism of ANN, we proposed using a generalized Euclidean distance where by changing its coefficients new formation of clusters of alternatives can be achieved. The algorithm is interactive and the results are independent of the initial set-up information. Some examples and computational results are presented.

Keywords: Clustering, grouping, multiple criteria, multi-objective optimization, ranking, supervised ANN, unsupervised ANN

1. Introduction

Cluster analysis is concerned with grouping of alternatives into homogeneous clusters based on certain features. The clustering of multiple criteria alternatives can bring out the following benefits: It decreases the set of alternatives—since the decision-maker may be interested in those alternatives with similar features and discard other alternatives. It provides a basis for more in-depth evaluation of alternatives—once a set of clustered alternatives are selected, then this set can be explored in more depth for analysis, selection, and implementation purposes. It may provide a basis for analyzing multiple criteria/multiple decision-makers problems while each decision-maker may cluster alternatives differently and hence clustering of alternatives may provide a basis for communication and facilitating the decision-making process. In case of selection of a group of alternatives, each decision-maker can be in charge of one clustered group; hence the best solution is selected from each clustered group by the designated decision-maker.

The need for more analytical and efficient methods for solving these problems, particularly for large problems has long been recognized. During the past two decades, many methods were developed for solving multi-criteria and clustering problems separately; however, there are very few works that have attempted to solve the multi-criteria clustering problem. Most suggested methods for clustering fall into the following categories: (1) criteria reduction clustering; (2) consensus clustering; (3) constrained clustering; and (4) direct clustering algorithms. Criteria reduction clustering reduces multi-criteria into a single criterion by combining all criteria. Clusters obtained by consensus clustering technique (Day, 1986a, 1986b) is by applying single criteria clustering algorithm for each criterion. Constrained clustering algorithms (Lefkovitch, 1985; Ferligoj and Batagelj, 1982, 1983; Ferligoj and Lapajne, 1987) consider a particular criterion as the clustering criterion and all other criteria are reduced to constraints for the problem. Direct clustering algorithms are based on dynamic clusters (Hanani, 1979) using the concept of kernel, as a representation of any given criterion.

Decision-making involves generating and evaluating various alternatives and choosing the best course of action among various alternatives. In nearly all decision-making problems, several conflicting criteria for judging and evaluating various alternatives exist. The main concern of the decision-maker is to fulfill his/her goals while satisfying the constraints of the system. For good overview of and references on multiple criteria decision making (MCDM) problems, see Dyer *et al.* (1992), Goicoechea *et al.* (1982), Hwang and Yoon (1981), and Steuer (1986).

In traditional MCDM or multiple objective optimization (MOO) methods, the entire set of alternatives are considered for the selection of the best alternative by one decision-maker. Such a selection is usually made on the basis of certain principles, accepted practices, or heuristics, and the method of selection and the decision-maker remain the same in treating the entire set of alternatives. However, in many decision-making situations, all alternatives cannot be treated the same way, using the same decision-maker and the same method of selection. Furthermore, a number of best alternatives (instead of one best alternative) may need to be selected. In these cases, the clustering or grouping of alternatives becomes necessary. Consider the example of selecting computer systems for a large corporation. Different computer systems are in use today based on their function and computing power. In general, computers can be divided as desktop machines, network file servers and database servers. Each category requires different configuration of computers. Clustering can help a single or multiple decision-makers classify alternatives into the above three categories and carry out selection from these groups.

Clustering has also been extensively treated in the field of pattern recognition and neural networks. Various clustering and neural network algorithms exist for the same purpose. In pattern recognition, we have maximin-distance algorithm which is based on Euclidean distance concept, K-means algorithm which is based on the minimization of performance index, which is defined as the sum of the squared distances from all points in a cluster domain to cluster center. The behavior of this algorithm is influenced by the number of cluster centers specified, the choice of initial cluster centers, the order in which the samples are taken, and the geometrical properties of the data. Isodata algorithm, which is similar in principle to the K-means procedure in the sense that cluster centers are iteratively determined, however Isodata represents a fairly comprehensive set of additional heuristic procedures which have been incorporated into an iterative scheme. In neural networks, there are back propagation network, counter propagation network, learning vector quantizer, Kohonen network, ART 1 & 2 and a host of others. Some of them utilizing supervised learning and others using unsupervised learning method. Although there are plenty of networks and algorithms, we have not seen much by way of application of these networks and algorithms for MCDM as we have chosen to do so. Clustering has been applied in many areas, such as biology, data recognition, medicine, pattern recognition, production flow analysis, task selection, control engineering, automated systems, and expert systems. In this paper, we utilize a variable-parameter, self-organizing neural network clustering system to solve the multi-criteria clustering problem.

Our motivation to solve the above MCDM problem using artificial neural network (ANN), is due to the enormous success ANN has enjoyed in pattern recognition studies, in particular pattern classification. ANN is a learning paradigm. This learning property can be used to train ANN to perform the classification to the required degree of accuracy. ANN can be used to classify (form clusters of) discrete alternatives. The unsupervised ANN is particularly good at forming clusters of alternatives in solution space, when provided with a threshold. Another area where ANN is adept is mapping of input to output. ANN is capable of learning the complex relationship between the input and output parameters.

In our approach, we attempt to use the above advantages of ANN to carry out the classification task and finally selection by capturing decision-maker's preference structure. Decision-maker in most cases will have some sort of intuition about the alternative distribution: This could be used as a starting point for cluster formation in an interactive session. Clustering is grouping alternatives that have some common property or meet a common objective. If an alternative meets a set of objectives, alternatives in the close neighborhood are also more than likely to meet the same set of objectives. A threshold defines this neighborhood. By grouping alternatives in clusters, we reduce the solution space to a manageable number of alternatives. The paradigm of clusters helps in doing just that.

A multi-criteria alternative belongs to a particular class (cluster), if it is closer in distance to that particular class than to any other class. The distance function is used as a classification tool, since points that are closer together in Euclidean space are similar. This method provides good clusters, when alternatives exhibit clustering properties. A few alternatives may fall outside of all the classes. These alternatives represent solutions that are far removed from characteristics represented by the existing classes. These are called exceptional or odd alternatives, and we show an approach to identify them.

In Section 2, basic definitions and methods for multi-criteria clustering are introduced. In Section 3, a review of self-organizing neural network is given. In Section 4, we present our approach for multi-criteria unsupervised learning. We first present an optimization problem, and then we propose our variable-parameter unsupervised learning algorithm that uses a generalized Euclidean distance measure and a momentum term in the weight vector for updating equations for forming clusters. Section 5 results of some computational analysis with the clustering method is presented. In Section 5, we also discuss the effect of inaccuracy (random mistakes), sensitive analysis, and the effect of odd alternatives. Section 6 is the approach for the selection of the best multi-criteria alternative using feedforward artificial neural networks. Section 7 is the conclusion. Several examples are given.

2. A naïve approach for multi-criteria clustering

In this section we present a simple and naïve approach for clustering multi-criteria alternatives for the purpose of illustration of the complexity of the problem. In Section 4, we propose a more complex approach that can very effectively and without the involvement of the decision-maker can cluster alternatives.

Two well-known methods of array-based clustering applied to group technology problems are rank order clustering (King, 1980b) and direct clustering analysis (Chan and Milner, 1982). Rank order clustering is a technique for block diagnolization by repeatedly rearranging the columns and rows of the objective-alternative incidence matrix $\{a_{ij}\}$. The matrix $\{a_{ij}\}$ is considered as m binary numbers, with each row representing a single binary number (0 or 1). This method has two disadvantages. First, the quality of the results is strongly dependent on the initial disposition of the objective-alternative incidence matrix. Second, the binary value (a power of 2) that is used for the rearrangement restricts the problem size.

Directive clustering analysis rearranges the rows with the left-most positive element to the top and the columns with the top-most positive element to the left of the incidence matrix (Chan and Milner, 1982). After several iterations, all the positive elements will form diagonal blocks from the top left corner to the bottom right corner. Array-based clustering has other disadvantages (Kusiak, 1987; Kusiak and Heragu, 1987). Also see (Kusiak, 1987; Kusiak and Heragu, 1987) for an integer programming formulation of the clustering problem, known as the p-median model.

2.1. *The naïve multi-criteria clustering approach using rank order clustering*

We propose using the following approach where instead of the binary values for each elements of objective-alternative incidence matrix $\{a_{ij}\}$, a real number is used. This value represents a numerical value of a given criterion. Our clustering problem is then to group or cluster alternatives into exclusive subsets. In the following approach each column represents one alternative, and each row represent the values for criteria associated with that alternative.

Algorithm

Step 1 Consider objectives and alternatives in an incidence matrix form.

Step 2 Normalize the values of each row of the matrix between 0 and 1 (or alternatively set minimum and maximum values outside of the existing range, and then normalize the values). This convert alternatives/criteria values to comparable units.

Step 3 The value of ith row is obtained by $a_{in}x2^0 + a_{i(n-1)}x2^1 + \cdots + a_{i(n-r)}x2^r + \cdots + a_{i1} x2^{n-1}$.

Step 4 Rearrange each rows in decreasing order.

Step 5 Repeat the procedure for finding the value of columns and rearrange them.

Step 6 At this point, alternatives can be grouped by choosing the number of families and grouping adjacent alternatives into the same group.

Step 7 Additional improvement can be obtained by visual inspections and rearranging some of the alternatives.

Example 1

Step 1 Consider the following tri-criteria problem with four alternatives—we wish to cluster them to two or three clusters.

	a_1	a_2	a_3	a_4
f_1	2	9	1	5
f_2	3	7	3	4
f_3	10	4	8	3

Step 2 Normalize the values between 0–1 (assuming 0 is the lowest value and 10 is the highest value) by dividing all the elements with the highest in the matrix.

	a_1	a_2	a_3	a_4	
f_1	0.2	0.9	0.1	0.5	2.66
f_2	0.3	0.7	0.3	0.4	6.20
f_3	1	0.4	0.8	0.3	11.5

Step 3 Calculate the values of the rows—

$f_1 = 8 \times 0.2 + 4 \times 0.9 + 2 \times 0.1 + 1 \times 0.5 = 2.66$
$f_2 = 8 \times 0.3 + 4 \times 0.7 + 2 \times 0.3 + 1 \times 0.4 = 6.20$
$f_3 = 8 \times 1 + 4 \times 0.4 + 2 \times 0.8 + 1 \times 0.3 = 11.5$

Step 4 Rearrange each row by decreasing order (3,2,1)

	a_1	a_2	a_3	a_4
f_3	1	0.4	0.8	0.3
f_2	0.3	0.7	0.3	0.4
f_1	0.2	0.9	0.1	0.5
	4.8	3.9	3.9	2.5

Step 5 Similarly the values of the columns are (4.8, 3.9, 3.9, 2.5).

Step 6 Now, alternatives 1 and 2 can be clustered in one family and alternative 3 and 4 in the second family.

Step 7 By visual inspection, we can rearrange columns 2 and 3 (they also have equal column values), to have a better clustering.

	a_1	a_2	a_3	a_4
f	1	0.8	0.4	0.3
f_2	0.3	0.3	0.7	0.4
f_1	0.2	0.1	0.9	0.5

Alternatives 1 and 3 belong to Family 1. Alternatives 2 and 4 belong to Family 2. Although, the above method is simple, it has two disadvantages, (1) its final solution may dependent on the initial or starting solution, and (2) the final matrix should be visually inspected to group the alternatives manually. The approach proposed in Section 4 overcome both these problems.

3. Review of neural networks

An artificial neural network is mathematical modeling of biologically motivated computation believed to occur in the human brain. It is designed to exploit the massively parallel local processing and distributed representation capability.

A neural network is a highly parallel computation system, modeled after the human brain. Neural networks are especially powerful for identifying patterns, trends, and internal relationships. Common applications of neural networks are as follows: signal processing, pattern recognition, fault diagnosis, decision-making and analysis, data clustering and classification, and system identification.

A neural network consists of simple processing elements (nodes), connection links and learning rules. Each processing element collects inputs from multiple sources, and produces an output after the weighted combined inputs are processed by an activation function. Connections are the links between nodes. The connections are characterized individually by their linkage strength or weights, and collectively by their configuration. Common configurations of neural networks are full-interconnection and feed-forward. For more information on artificial neural networks see (Rumelhart and Zipser, 1985; Rumelhart *et al.*, 1986; Kohonen, 1984, 1988; Grossberg, 1976, 1987; Kamal, 1996).

There are two main types of learning: supervised and unsupervised learning. For supervised learning, the training pattern set consists of typical patterns whose class identities are known (for examples and references see Malakooti and Zhou, 1994, 1995; Wang and Malakooti, 1992). For unsupervised learning, information about the class membership or label for the training patterns is not given because of either lack of knowledge or the high cost of providing the class labels associated with each of the training patterns.

A vast amount of effort has already been dedicated to the study of supervised learning algorithms, such as the Boltzman machine, the back-propagation algorithm (Rumelhart *et al.*, 1986), high-order networks, GMDH networks, radial basis function networks. Less attention has been devoted to unsupervised learning algorithms, which do not require explicit tutoring by input–output correlation and which spontaneously self-organize upon presentation of input patterns.

Our approach for clustering multicriteria alternatives is based on unsupervised neural networks. Two well-known unsupervised learning neural network models are competitive learning (Grossberg, 1976, 1987; Kohonen, 1984, 1988; Rumelhart and Zipser, 1985) and self-organizing maps (Kohonen, 1984). The clustering process of unsupervised learning neural networks is to find the internal representation of data patterns set without knowing the class labels of training patterns, i.e., to identify several prototypes or exemplars that can serve as cluster centers.

4. Our approach for multi-criteria clustering using unsupervised learning algorithm

In this section, we formulate the problem as an optimization problem and then develop a heuristic-based ANN approach to solve the problem. Our neural network approach takes advantage of both methods discussed in Section 3. In addition, we develop and use a generalized distance measurement.

For the clustering problem, the inputs to neural networks are the set of all alternatives. Each alternative, presented by an n-tuple vector $\boldsymbol{a}_i$ is one input, the set of all multi-criteria alternatives are $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m$, that is we have m different inputs.

4.1. *Problem formulation*

We denote the n-tuple multi-criteria alternatives as $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m$, and the number of clusters as $\boldsymbol{R}$; where $\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_m$ are given and $\boldsymbol{R}$ is unknown. For each cluster r, $r = 1, 2, \ldots, \boldsymbol{R}$, we need to find it's cluster center vectors: $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_R$. We define the dissimilarity measurement $d(\boldsymbol{a}_s, \boldsymbol{x}_r)$ for each pair of a multi-criteria alternative $\boldsymbol{a}_s$ and a cluster center vector $\boldsymbol{x}_r$ by the following generalized Euclidean distance:

$$d(\boldsymbol{a}_s, \boldsymbol{x}_r) = \left(\sum_{i=1}^{n} k_i (a_{si} - x_{ri})^2 \right)^{1/2}$$

where $k_1, k_2, \ldots, k_n$ are coefficients which are unknown and should be assessed, and $k_1 + k_2 + \cdots + k_n = 1$ and $0 \leq k \leq 1$, for $i = 1, 2, \ldots, n$. In general, the generalized Euclidean distance can represent more practical problems than the Euclidean distance. When solving the cluster formation problem, the coefficients in the generalized Euclidean distance can be used to represent the importance index to each attribute in the multicriteria alternatives.

A main difference between our approach of multi-criteria clustering and the general clustering method lies in the fact that we deal with criteria that have to be optimized and our approach for selection of the best alternative or alternatives (in the next section) completes this task. Furthermore, parameters k

introduced in the above equation represent the assessment of the relevance importance of the criteria.

We define clustering efficiency (CE) as 1 minus of the ratio of number of exceptional alternatives (NE) to the total number of alternatives (N).

$$PE = (1 - NE/N) * 100$$

The quality of a clustering method is denoted by this measure. Higher the value of CE, better the clustering method is.

We formulate the problem as an optimization problem to find the cluster centers. The objective is to minimize the total dissimilarity among multi-criteria alternatives for all $\boldsymbol{R}$ clusters and maximize clustering efficiency.

Problem: Bicriteria for clustering

$$\min F(d) = \sum_{r=1}^{R}\sum_{s=1}^{m} y_{sr}\left(\sum_{i=1}^{n} k_1(a_{xi} - x_{ri})^2\right)^{1/2} \quad (4.1)$$

$$\max CE \quad (4.2)$$

$$\sum_{i=1}^{n} k_t = 1 \text{ where } 0 \le k_i \le 1, \quad \forall i = 1,2,\ldots,n \quad (4.3)$$

$$\text{s.t.} \quad \sum_{r=1}^{R} y_{sr} = 1 \text{ for } s = 1,2,\ldots,m \quad (4.4)$$

where

$$y_{sr} = 1 \text{ if } d(\boldsymbol{a}_s, \boldsymbol{x}_r) \le d(\boldsymbol{a}_s, \boldsymbol{x}_p),$$
$$p = 1,2,\ldots,R, p \ne r$$
$$y_{sr} = 0 \text{ otherwise}$$

Constraint (4.4) guarantees that one alternative is assigned to one and only one cluster. Problem 1 is a nonlinear quadratic with mixed integer variables. It is very difficult to find an optimal solution to this problem by existing optimization techniques when the problem size increases. Therefore, we develop our own heuristic iterative procedure that uses an unsupervised learning neural network to solve this problem. We develop an unsupervised learning algorithm based on the competitive learning that has a simple structure and a computational procedure. We modify the competitive learning algorithm by using the generalized Euclidean distance as the dissimilarity measure and adding a momentum term in the weight vector updating equations to improve stability. The solution can be used when the number of clusters is known as in certain problems as well as when it is unknown.

4.2. *Momentum term*

We add a momentum term in the weight vector updating equation. A large learning rate, β, corresponds to rapid learning but might also result in oscillations. The use of a momentum term specifies that changes in $\boldsymbol{w}_r$ at the (t)th step should be somewhat similar to the changes undertaken at the $(t-1)$th step. In this way, it keeps the algorithm from oscillating. The modified weight vector updating equation is

$$\Delta \boldsymbol{w}_r(t) = \beta(t)(\boldsymbol{a}_s - \boldsymbol{w}_r(t-1)) + (1 - \beta(t))\, \Delta \boldsymbol{w}_r(t-1)$$

where β is the learning rate, and $0 \le \beta(t) \le 1$.

4.3. *Summary of the developed algorithm*

The procedure consists of two phases. In phase I, the cluster centers are identified. Phase I is applicable when number of clusters $\boldsymbol{R}$ is either known or unknown. In phase II, clustering of alternatives is performed based on the distances between multi-criteria alternatives and the cluster centers.

4.3.1. *Define*

k_i—Coefficients in the generalized Euclidean formula
$\boldsymbol{R}$—Number of clusters (number of output nodes for the network)
p—Number of alternatives (number of input patterns)
$\boldsymbol{w}_r(t)$—Weight vectors (cluster center vector)
$\boldsymbol{a}_s$—Input vector (multicriteria alternative)
Ed_r—Euclidean distance between weight vector $\boldsymbol{w}_r(t)$ and input vector $\boldsymbol{a}_s$
$\beta(t)$—Learning rate
δ—Stop parameter
t—Training time index (iteration index)
g_r—Number of alternatives in cluster r
μ—Threshold for clustering.

4.3.2. *Phase I*

To find the cell center when $\boldsymbol{R}$ is given

Step 1 Set $k_i, \beta, \delta, p, \boldsymbol{R}$.
Step 2 Set $t = 1$ Generate initial weight vectors $w_r(1); \Delta w_r(1) = 0, r = 1,2,\ldots,\boldsymbol{R}$ Set $s = 1$
Step 3 Present input pattern $\boldsymbol{a}_s$

Step 4 Compute the generalized distance between input pattern a_s and all weight vectors

$$Ed_r = ||\boldsymbol{a}_s - \boldsymbol{w}_r(t)||k^2 = k_1(a_{s1} - w_{r1}(t))^2 + k_2(a_{s2} - w_{r2}(t))^2 \ldots + k_n(a_{sn} - w_{rn}(t))^2 \ldots \text{ for } r = 1, 2, \ldots, \boldsymbol{R}$$

Step 5 Find the least distance $Ed_r = \min\{Ed_r\}$, where $r = 1, 2, \ldots, \boldsymbol{R}$.

Step 6 Since input pattern $\boldsymbol{a}_s$ belonging to cluster r^*, update $\Delta\boldsymbol{w}_{r*}(t+1) = \beta(\boldsymbol{a}_s - \boldsymbol{w}_{r*}(t)) + (1-\beta)\Delta\boldsymbol{w}_{r*}(t)$

$$\Delta\boldsymbol{w}_r(t+1) = \Delta\boldsymbol{w}_r(t) \text{ for } r = 1, 2, \ldots, \boldsymbol{R} \text{ and } r \neq r^*$$

$$\boldsymbol{w}_r(t+1) = \boldsymbol{w}_r(t) + \Delta\boldsymbol{w}_r(t+1) \text{ for } r = 1, 2, \ldots, \boldsymbol{R}.$$

Step 7 Set $s = s+1$. If $s < p$, $t = t+1$, go to Step 3. Otherwise go to Step 8.

Step 8 $s = p$ If $\Delta\boldsymbol{w}_r(t+1) < \delta$ for $r = 1, 2, \ldots, \boldsymbol{R}$, stop. Otherwise, set $s = 1$, $t = t+1$, and go to Step 3.

To find the cell center when ***R*** *is unknown*

Step 1 Set $k_i, \beta, \delta, p, \mu$.

Step 2 Set $t = 1$, $r = \boldsymbol{R} = 1$. Generate initial weight vectors $\boldsymbol{w}_r(1)$; $\Delta\boldsymbol{w}_r(1) = 0$ Set $s = 1$.

Step 3 Present input pattern $\boldsymbol{a}_s$.

Step 4 Compute the generalized distance between input pattern $\boldsymbol{a}_s$ and all weight vectors.

$$Ed_r = ||\boldsymbol{a}_s - \boldsymbol{w}_r(t)k^2 = k_1(a_{s1} - w_{r1}(t))^2 + k_2(a_{s2} - w_{r2}(t))^2 \ldots + k_n(a_{sn} - w_{rn}(t))^2 \text{ for } r = 1, 2, \ldots, \boldsymbol{R}$$

Step 5 Find the least distance $Ed_r* = \min\{Ed_r\}$, where $r = 1, 2, \ldots, \boldsymbol{R}$.

Step 6 If $Ed_r * \ \mu$, then input pattern $\boldsymbol{a}_s$ belonging to center r^*, update $\Delta\boldsymbol{w}_{r*}(t+1) = \beta(\boldsymbol{a}_s - \boldsymbol{w}_{r*}(t)) + (1-\beta)\Delta\boldsymbol{w}_{r*}(t)$ $\Delta\boldsymbol{w}_r(t+1) = \Delta\boldsymbol{w}_r(t)$ for $r = 1, 2, \ldots, \boldsymbol{R}$ and $r \neq r^*$ $\boldsymbol{w}_r(t+1) = \boldsymbol{w}_r(t) + \Delta\boldsymbol{w}_r(t+1)$for $r = 1, 2, \ldots, \boldsymbol{R}$. Otherwise, if $Ed_r* > \mu$, then start another cluster r. Update $r = r+1$, $\boldsymbol{R} = \boldsymbol{R}+1$. Set $\boldsymbol{w}_r(t) = \boldsymbol{a}_s$.

Step 7 Set $s = s+1$. If $s < p$, $t = t+1$, go to Step 3. Otherwise go to Step 8.

Step 8 $s = p$.
If $\Delta\boldsymbol{w}_r(t+1) < \delta$ for $r = 1, 2, \ldots, \boldsymbol{R}$, stop. Otherwise, set $s = 1$, $t = t+1$, and go to Step 3.

4.3.3. *Phase II. To cluster p multicriteria alternatives into R clusters*

Step 1 Set $s = 1$. Set $g_r = 0$, for $r = 1, 2, \ldots, \boldsymbol{R}$.

Step 2 Multicriteria alternative a_s belongs to cluster r if

$$d(\boldsymbol{a}_s, \boldsymbol{x}_r) \leq d(\boldsymbol{a}_s, \boldsymbol{x}_m), m = 1, 2, \ldots, \boldsymbol{R}, m \neq r$$

$$\text{Set } g_r = g_r + 1$$

Step 3 If $s = p$, Stop. Otherwise set $s = s+1$, go to Step 2.

4.4. *Clustering application examples*

In this section, we solve two examples by variable-parameter unsupervised learning algorithm.

Example 2
For example consider buying a set of computers for a company that vary in terms of their reliability and computing powers. Let us suppose the set of six computers are considered, whose ratings of are from 0 to 6. For example, in Table 1, alternative 1 has the lowest reliability and the highest computing power.

Table 1. Depicts multicriteria alternative used in clustering example below

#	*Alternative*	#	*Alternative*
a_1	(1.5, 5.5)	a_4	(4, 2)
a_2	(3.8, 3)	a_5	(2, 5)
a_3	(1.6, 4.9)	a_6	(4.5, 4)

The method clusters alternatives 1, 3, and 5 into cluster number 1; and clusters alternatives 2, 4 and 6 into cluster number 2. See Appendix 1 for the details of the algorithm.

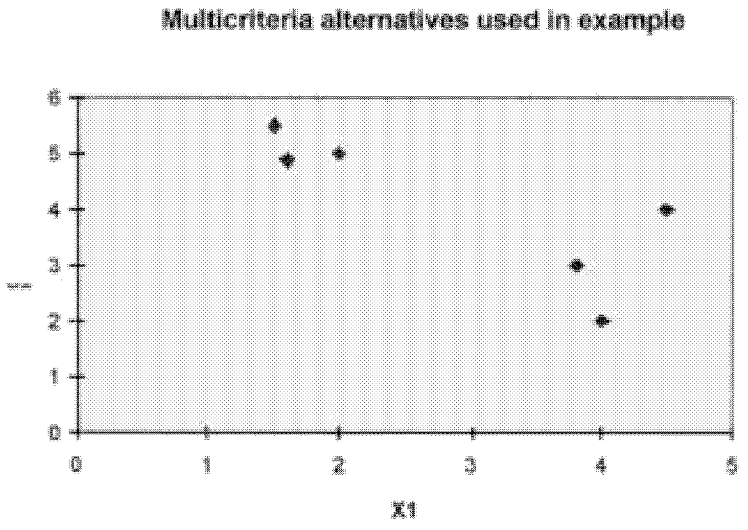
Example 3
In this example 2×100 alternatives are randomly generated (see Table 2) and the multi-criteria clustering algorithm is used to cluster alternatives,

Table 2. Presents 2 attribute and 100 alternatives and used in clustering example below. After using the algorithm six clusters of alternatives are generated, see Table 3

Alt #	*Alternative*	*Alt #*	*Alternative*	*Alt #*	*Alternative*	*Alt #*	*Alternative*
a_1	(0.77, 0.55)	a_{26}	(0.97, 0.60)	a_{51}	(0.47, 0.05)	a_{76}	(0.69, 0.95)
a_2	(0.77, 0.89)	a_{27}	(0.31, 0.17)	a_{52}	(0.64, 0.89)	a_{77}	(0.95, 0.34)
a_3	(0.77, 0.84)	a_{28}	(0.26, 0.81)	a_{53}	(0.12, 0.40)	a_{78}	(0.12, 0.22)
a_4	(0.59, 0.50)	a_{29}	(0.11, 0.49)	a_{54}	(0.31, 0.01)	a_{79}	(0.33, 0.69)
a_5	(0.28, 0.45)	a_{30}	(0.70, 0.22)	a_{55}	(0.38, 0.75)	a_{80}	(0.41, 0.66)
a_6	(0.55, 0.58)	a_{31}	(0.18, 0.34)	a_{56}	(0.29, 0.20)	a_{81}	(0.23, 0.11)
a_7	(0.00, 0.52)	a_{32}	(0.25, 0.72)	a_{57}	(0.88, 0.65)	a_{82}	(0.98, 0.21)
a_8	(0.39, 0.19)	a_{33}	(0.10, 0.17)	a_{58}	(0.71, 0.87)	a_{83}	(0.26, 0.80)
a_9	(0.33, 0.30)	a_{34}	(0.01, 0.43)	a_{59}	(0.16, 0.95)	a_{84}	(0.42, 0.00)
a_{10}	(0.87, 0.48)	a_{35}	(0.79, 0.33)	a_{60}	(0.05, 0.95)	a_{85}	(0.15, 0.11)
a_{11}	(0.32, 0.26)	a_{36}	(0.64, 0.73)	a_{61}	(0.44, 0.07)	a_{86}	(0.90, 0.11)
a_{12}	(0.64, 0.45)	a_{37}	(0.32, 0.13)	a_{62}	(0.64, 0.21)	a_{87}	(0.05, 0.05)
a_{13}	(0.07, 0.98)	a_{38}	(0.03, 0.67)	a_{63}	(0.39, 0.69)	a_{88}	(0.08, 0.17)
a_{14}	(0.35, 0.40)	a_{39}	(0.71, 0.23)	a_{64}	(0.57, 0.95)	a_{89}	(0.16, 0.30)
a_{15}	(0.62, 0.61)	a_{40}	(0.84, 0.03)	a_{65}	(0.18, 0.47)	a_{90}	(0.80, 0.45)
a_{16}	(0.94, 0.09)	a_{41}	(0.59, 0.46)	a_{66}	(0.12, 0.70)	a_{91}	(0.35, 0.71)
a_{17}	(0.56, 0.14)	a_{42}	(0.02, 0.74)	a_{67}	(0.88, 0.27)	a_{92}	(0.86, 0.05)
a_{18}	(0.98, 0.16)	a_{43}	(0.51, 0.05)	a_{68}	(0.47, 0.57)	a_{93}	(0.76, 0.48)
a_{19}	(0.04, 0.62)	a_{44}	(0.22, 0.60)	a_{69}	(0.50, 0.91)	a_{94}	(0.61, 0.98)
a_{20}	(0.79, 0.03)	a_{45}	(0.90, 0.41)	a_{70}	(0.99, 0.53)	a_{95}	(0.49, 0.33)
a_{21}	(0.12, 0.61)	a_{46}	(0.42, 0.14)	a_{71}	(0.21, 0.12)	a_{96}	(0.02, 0.88)
a_{22}	(0.64, 0.03)	a_{47}	(0.51, 0.90)	a_{72}	(0.35, 0.33)	a_{97}	(0.82, 0.73)
a_{23}	(0.02, 0.34)	a_{48}	(0.87, 0.12)	a_{73}	(0.23, 0.34)	a_{98}	(0.97, 0.50)
a_{24}	(0.40, 0.08)	a_{49}	(0.76, 0.96)	a_{74}	(0.27, 0.59)	a_{99}	(0.22, 0.58)
a_{25}	(0.46, 0.93)	a_{50}	(0.30, 0.91)	a_{75}	(0.04, 0.29)	a_{100}	(0.70, 0.91)

Table 3. Presents a table of clusters along with alternatives belonging to the cluster

Cluster #	*Alternatives*	*Cluster centers*
1	(0.77, 0.55), (0.59, 0.50), (0.55. 0.58), (0.87, 0.48), (0.64, 0.45), (0.62, 0.61), (0.97, 0.60), (0.79, 0.33), (0.64, 0.73), (0.59, 0.46), (0.90, 0.41), (0.88, 0.65), (0.47, 0.57), (0.99, 0.53), (0.95, 0.34), (0.80, 0.45), (0.76, 0.48), (0.82, 0.73), (0.97, 0.50)	(0.77, 0.52)
2	(0.77, 0.89), (0.46, 0.93), (0.26, 0.81), (0.25, 0.72), (0.51, 0.90), (0.76, 0.96), (0.30, 0.91), (0.64, 0.89), (0.38, 0.75), (0.71, 0.87), (0.39, 0.69), (0.57, 0.95), (0.50, 0.91), (0.69, 0.95), (0.33, 0.69), (0.41, 0.66), (0.26, 0.8), (0.35, 0.71), (0.61, 0.98), (0.70, 0.91)	(0.49, 0.84)
3	(0.28, 0.45), (0.39, 0.19), (0.33, 0.30), (0.32, 0.26), (0.35, 0.40), (0.56, 0.14), (0.02, 0.34), (0.40, 0.08), (0.31, 0.17), (0.18, 0.34), (0.10, 0.17), (0.32, 0.13), (0.51, 0.05), (0.42, 0.14), (0.47, 0.05), (0.12, 0.40), (0.31, 0.01), (0.29, 0.20), (0.44, 0.07), (0.21, 0.12), (0.35, 0.33), (0.23, 0.34), (0.04, 0.29), (0.12, 0.22), (0.23, 0.11), (0.42, 0.0), (0.15, 0.11), (0.05, 0.05), (0.08, 0.17), (0.16, 0.30), (0.49, 0.33)	(0.28, 0.20)
4	(0.07, 0.84), (0.00, 0.52), (0.04, 0.62), (0.12, 0.61), (0.11, 0.49), (0.01, 0.43), (0.03, 0.67), (0.02, 0.74), (0.22, 0.60), (0.18, 0.47), (0.12, 0.70), (0.27, 0.59), (0.22, 0.58)	(0.11, 0.60)
5	(0.07, 0.98), (0.16, 0.95), (0.05, 0.95), (0.02, 0.88)	(0.08, 0.94)
6	(0.94, 0.09), (0.98, 0.16), (0.79, 0.03), (0.64, 0.03), (0.70, 0.22), (0.71, 0.23), (0.84, 0.03), (0.87, 0.12), (0.64, 0.21), (0.88, 0.27), (0.98, 0.21), (0.90, 0.11), (0.86, 0.05)	(0.83, 0.14)



Graph 1. Depicts the distribution of 2 criteria 6 alternatives used in clustering example 1.

they are clustered in six clusters, see Appendix 2 for details.

5. Some experimental analysis with multicriteria clustering algorithm; effect of inaccuracy, sensitivity analysis, and effect of odd alternatives

5.1. *Some computational analysis*

In this section, we demonstrate the applicability of the developed algorithm for multiple criteria clustering of alternatives. This algorithm was written in C and implemented under UNIX. The samples have been generated using the random number generator of the standard C library. The sample size varies from 25 to 300 for 2, 3, 4, 5, and 6 attribute problems.

The number of pareto optimal clusters varies according to the distribution of the alternatives in n-dimensional space. For a given number of attributes and number of alternatives, experiments were conducted to obtain the optimum cluster in terms of minimizing the generalized Euclidean distance (GED) for all alternatives with respect to their cluster centers and maximizing clustering efficiency.

GED is inversely proportional to the number of clusters. The higher the number of clusters, the lower is the GED. This essentially depends on the number of alternatives in each cluster. If alternatives get distributed in more number of clusters, then the GED reduces.

In general the number of clusters increases with the increase in the sample size and increase in sample attributes.

Table 4. Displays the results of the computational experiments for the self-organizing neural network using variable weights, for various sample sizes and number of attributes

Sample size	*25*			*50*			*100*			*200*			*300*			*Average CE*
No. of Attr.	*R*	GED	CE	*R*	GED	CE	*R*	GED	CE	*R*	GED	CE	*R*	GED	CE	
2	3	19.77	100	6	26	98	6	39.73	97	7	59.01	99.5	5	74.62	100	98.9
3	4	22.23	96	5	43.19	100	4	57.26	97	7	104.4	99	8	127.9	100	98.4
4	4	35.78	100	5	50.06	100	6	81.92	100	9	119.6	100	9	131.9	100	100
5	5	38.06	100	6	65.13	98	6	90.52	100	8	135.3	100	8	190.8	100	99.6
6	5	42.13	100	8	62.25	100	7	89.54	99	9	129.9	100	8	254.6	99	99.6

Where R, number of clusters; GED, generalized Euclidean distance measure; CE, Clustering efficiency.

Table 5. Displays the effect of inaccuracy percentage on the clustering algorithm using Example 1

Inaccuracy %	*Number of clusters R*	*Alternatives in each cluster*	*Selection accuracy %*
0	2	I-(1,3,5), II-(2,4,6)	100
5	2	I-(1,3,5), II-(2,4,6)	100
10	2	I-(1,3,5), II-(2,4,6)	100
15	2	I-(1,3,5), II-(2,4,6)	100
20	2	I-(1,3,5), II-(2,4,6)	100
25	2	I-(1,3,5), II-(2,4,6)	100

We define (CE) as 1 minus of the ratio of number of exceptional alternatives (NE) to the total number of alternatives.

$$PE = (1 - NE/N) * 100$$

CE in general is very high with the average being 99.3%. With the increase in the dimensionality of the multiple criteria alternative, the clustering efficiency changes very slightly. For the problems in the computational experiment, the average CE varies from 98.4% to 100%. The results of the computational experiments are summarized in Table 4.

The effect of inaccuracy on the clustering algorithm is presented in the section below. Consider the problem data in example 1 above. Table 5 presents final clustering arrangement with 0% inaccuracy, 5% inaccuracy, 10% inaccuracy, 15% inaccuracy, 20% inaccuracy, and 25% inaccuracy.

5.2. *Effect of inaccuracy*

The effect of inaccuracy on the sensitive of clustering algorithm is minimal. The selection accuracy is 100% for a inaccuracy percentage of 0, 5, 10, 15, 20, and 25. The selection accuracy (SA) is defined as the ratio of number of misclassified alternatives to the product of the number of clusters and total number of alternatives.

$$SA = \left(1 - \frac{N}{RM}\right) * 100$$

where N is the number of alternatives misclassified, R is the number of clusters and M is the total number of alternatives.

5.3. *Sensitivity analysis—effect of adding alternatives*

The clustering algorithm is sensitive to the effect of adding additional alternatives. Three possible outcomes can result—

- If the alternatives added are close (distance less than threshold value) to existing weight vectors, then the alternatives will be classified into one of the existing clusters. The weight vector for these clusters will be modified to reflect the addition of new alternatives.
- If the alternatives are at a distance greater than the permissible threshold value for all existing clusters, then additional clusters will be formed. These new alternatives belong to this cluster. The weight vectors of these clusters are also modified, if the clusters have more than one alternative.
- If the cluster centers were formed when their number is specified, then the effect of adding additional alternatives is to (a) add the alternatives to the nearest cluster and (b) update the weight vector of that cluster to reflect the new alternatives.

Example 4
Updating weight vector

#	*Alternative*	#	*Alternative*
a_1	(1.5, 5.5)	a_5	(2, 5)
a_2	(3.8, 3)	a_6	(4.5, 4)
a_3	(1.6, 4.9)	a_7	(1.9, 4.3)
a_4	(4, 2)		

Additional alternative a_7 has been added to the above example. This results in maintaining the same number of clusters but updating the weight vector. The variable weights for parameters in the examples are $k_1 = k_2 = 0.5$.

Cluster number	*Alternatives*	*Cluster center*
1	a_1, a_3, a_5, a_7	(1.75, 4.93)
2	a_2, a_4, a_6	(4.10, 3.0)

Example 5
Adding a new cluster center

#	*Alternative*	#	*Alternative*
a_1	(1.5, 5.5)	a_5	(2, 5)
a_2	(3.8, 3)	a_6	(4.5, 4)
a_3	(1.6, 4.9)	a_7	(1.0, 0.5)
a_4	(4, 2)		

a_7 has been added to the above example and this time the effect is of adding a new cluster, as this new alternative does not belong to the existing clusters. The variable weights for parameters in the example are $k_1 = k_2 = 0.5$.

Cluster number	*Alternatives*	*Cluster center*
1	a_1, a_3, a_5	(91.7, 5.13)
2	a_2, a_4, a_6	(4.10, 3.0)
3	a_7	(1.0, 0.5)

5.4. *Odd families*

During clustering, there might be some alternatives that are spatially far away from other cluster centers. These alternatives form clusters of their own. These alternatives have to be treated differently from others. These alternatives are called odd alternatives and the clusters they form are odd clusters. In Example 5, cluster 3 is an example of odd cluster.

Our self-organizing clustering algorithm does take such alternatives into consideration. These alternatives will be presented in their individual clusters with cluster properties. These odd alternatives have an impact on percentage of exceptional element measure.

6. Selection of MCDM alternatives using feedforward ANN

In this section, we consider discrete sets of alternatives with the assumption that there exists a multiple attribute utility function (MAUF) that can represent the decision-maker's preference structure. We use the theory of feedforward ANN for representing MAUFs (which in turn represents the decision-maker) to rank alternatives and choose the most desirable one. Additive, multiplicative, and multi-linear are among the well-known structures that have been used for MAUFs. The assumptions underlying these MAUF structures are however restrictive. Identifying, assessing, and validating such functions, their structures, and their parameters can be formidable tasks for both decision maker and the analyst. Some interactive MCDM methods skirt this problem by not assessing the utility function completely and some other do not assume the existence of such utility functions (see Steuer, 1986 for reference). In this paper, we turn to an ANN approach to solve discrete MCDM problems. ANN method is a versatile yet robust approach to the quantification and representation of the DMs preferences (see Malakooti and Zhou (1994)). For other pioneering work in this area see Sun *et al.* (1996) and Stam *et al.* (1996).

It offers several advantages over MAUF-based and interactive methods that assume and use utility functions. First, it does not assume and need not verify that the utility function is of any particular structure or property, whereas the aforementioned methods cannot be relied on to yield satisfactory results if the utility function is, for example, highly nonlinear. Second, whereas the aforementioned interactive methods assess the utility function only partially, our ANN method generates a completely assessed function. Third, our ANN method can adjust and improve its representation as more information from the DM becomes available. Selection process can be divided into two steps—

(a) Selection of the best alternative or alternatives in each cluster

(b) Selection of the best alternative or alternatives from the selected alternatives in (a).

The selection process is carried out interactively with inputs from the decision-maker. The results from clustering from the previous section are presented to the decision-maker. Decision-maker chooses a cluster based on cluster characteristics. Decision-maker is then asked a number of questions. A feedforward neural network is trained on the decision-maker's response and the trained network chooses the best alternative. ANN is used in simulating the responses of the decision maker.

6.1. *Basics of feedforward ANN*

In general, a feedforward ANN consists of a set of nodes arranged in layers. The outputs of nodes in one layer are sent to input of nodes in another layer through links that increase or decrease outputs through weighting factors. Except for the input layer nodes, the net input to each node is the sum of the weighted outputs of the nodes in the previous layer. Each node is activated in accordance with the input to the node, the activation function of the node, and the bias of the node. In this paper, we use the approach presented by Malakooti and Zhou (1994) for solving this problem, however, the traditional feedforward ANN presented by Rumelhart *et al.* (1986) or other extension of it can be used without any loss of generality (see e.g., Pao, 1989).

6.2. *Cluster characteristics*

The clusters are obtained in the previous section. The minimum and maximum value of each objective function on all the clusters, along with the mean vector or weight vector and cluster variance are presented to the decision-maker in the form of a table. The decision-maker can pick one or more or all of the clusters presented. Each cluster represents different spheres of decision-making. One or more decision-makers may be involved in selecting one or more alternatives from the chosen cluster(s). The decision-maker's preference structure will vary for different clusters and different decision-maker may have a different preference structure for the same cluster.

Appendix 1: Details of Example 2 for clustering

Using variable parameter clustering algorithm for the example

Phase I:

Iteration 1

Step 1 Set $k_1 = k_2 = \cdots = k_n = 1, \beta = 0.5, \partial = 0.5, \mu = 3, p = 6$

Step 2 $w_1(1) = a_1 = [1.5, 5.5], s = 1; \Delta w_i(1) = 0$, for $i = 1, 2, 3, \ldots, 6$

Step 3 $t = 2$, Present $a_2 = [3.8, 3]$

Step 4 Compute $Ed_1 = ||a_2 - w_1(1)||^2 = 3.39$

Step 5 Set $Ed_1* = Ed_1 = 3.39$

Step 6 $Ed_1* > \mu = 3$
$s = s + 1 = 2$, produce a new center $w_2(2) = a_2 = [3.8, 3]$
$\Delta w_1(2) = \Delta w_2(2) = \cdots = \Delta w_6(2) = [0, 0]$
$w_1(2) = w_1(1) + \Delta w_1(2) = [1.5, 5.5]$

Step 7 $t = 2 < p = 6$, go to step 3

Iteration 2

Step 3 $t = 3$. Present $a_3 = [1.6, 4.9]$

Step 4 Compute $Ed_1 = ||a_3 - w_1(2)||^2 = 0.608$
$Ed_2 = ||a_3 - w_2(2)||^2 = 2.90$

Step 5 $Ed_1* = Ed_1 = 0.608$

Step 6 $Ed_1 = 0.608 < \mu = 3$ so a_3 belongs to cluster 1
$\Delta w_1(3) = 0.5(a_3 - w_1(2)) + 0.5\Delta w_1(2) = [0.05, -0.3]$
$\Delta w_2(3) = (0, 0)$
$w_1(3) = w_1(2) + \Delta w_1(3) = [1.55, 5.2]$
$w_2(3) = w_2(2) = [3.8, 3]$

Step 7 $t = 3 < p = 6$, go to step 3

Iteration 3

Step 3 $t = 4$. Present $a_4 = [4, 2]$

Step 4 Compute $Ed_1 = ||a_4 - w_1(3)||^2 = 4.11$
$Ed_2 = ||a_4 - w_2(3)||^2 = 1.01$

Step 5 $Ed_1* = Ed_2 = 1.01$

Step 6 $Ed_2 = 1.01 < \mu = 3$ so a_4 belongs to cluster 2
$\Delta w_2(4) = 0.5(a_4 - w_2(3)) + 0.5\Delta w_2(3) = [0.1, -0.5]$
$\Delta w_1(4) = (0, 0)$
$w_2(4) = w_2(3) + \Delta w_2(4) = [3.9, 2.5]$
$w_1(4) = w_1(3) = [1.55, 5.2]$

Step 7 $t = 4 < p = 6$, go to step 3

Iteration 4

Step 3 $t = 5$. Present $a_5 = [2, 5]$

Step 4 Compute $Ed_1 = ||a_5 - w_1(4)||^2 = 0.492$
$Ed_2 = ||a_5 - w_2(4)||^2 = 3.14$

Step 5 $Ed_1* = Ed_1 = 0.492$

Step 6 $Ed_1 = 0.492\mu = 3$ so a_5 belongs to cluster 1
$\Delta w_1(5) = 0.5(a_5 - w_1(4)) + 0.5\Delta w_1(4) = [0.225, -0.1]$
$\Delta w_2(5) = (0, 0)$
$w_1(5) = w_1(4) + \Delta w_1(5) = [1.775, 5.1]$.
$w_2(5) = w_2(4) = [3.9, 2.5]$

Step 7 $t = 5 < p = 6$, go to step 3

Iteration 5

Step 3 $t = 6$. Present $a_6 = [4.5, 4]$

Step 4 Compute $Ed_1 = ||a_6 - w_1(5)||^2 = 2.93$
$Ed_2 = ||a_6 - w_2(5)||^2 = 1.615$

Step 5 $Ed_1* = Ed_2 = 1.615$
Step 6 $Ed_2 = 1.615 \mu = 3$ so a_5 belongs to cluster 1
$\Delta w_2(6) = 0.5(a_6 - w_2(5)) + 0.5\Delta w_2(5) = [0.3, 0.75]$
$\Delta w_1(6) = (0, 0)$
$w_2(6) = w_2(5) + \Delta w_2(6) = [4.2, 3.25]$
$w_1(6) = w_1(5) = [1.775, 5.1]$
Step 7 $t = 6 = p = 6$, Stop.

Phase II
Iteration 1
Step 1 Set $s = 1$. *Set* $g_r = 0$, for $r = 1, 2, \ldots, \boldsymbol{R}$
Step 2 Present multicriteria alternative a_1 to cluster $r = 1, 2$
$d(\boldsymbol{a}_1, \boldsymbol{w}_1) = 0.485$ and $d(\boldsymbol{a}_1, \boldsymbol{w}_2) = 3.84$, hence a_1 belongs to cluster 1
Set $g_r = 1$.
Step 3 If $s = 1 < p = 6$, Set $s = s + 1 = 2$, go to Step 2.

Iteration 2
Step 2 Present multi-criteria alternative a_2 to cluster $r = 1$, 2
$d(\boldsymbol{a}_2, \boldsymbol{w}_1) = 2.91$ and $d(\boldsymbol{a}_2, \boldsymbol{w}_2) = 0.509$, hence a_2 belongs to cluster 2
Set $g_r = 2$.
Step 3 If $s = 2 < p = 6$, set $s = s + 1 = 3$, go to Step 2.

Iteration 3
Step 2 Present multi-criteria alternative a_3 to cluster $r = 1$, 2
$d(\boldsymbol{a}_3, \boldsymbol{w}_1) = 0.265$ and $d(\boldsymbol{a}_3, \boldsymbol{w}_2) = 3.32$, hence a_3 belongs to cluster 1
Set $g_r = 3$.
Step 3 If $s = 3 < p = 6$, set $s = s + 1 = 4$, go to Step 2.

Iteration 4
Step 2 Present multi-criteria alternative a_4 to cluster $r = 1$, 2
$d(\boldsymbol{a}_4, \boldsymbol{w}_1) = 3.81$ and $d(\boldsymbol{a}_4, \boldsymbol{w}_2) = 0.509$, hence a_4 belongs to cluster 2
Set $g_r = 4$.
Step 3 If $s = 4 < p = 6$, set $s = s + 1 = 5$, go to Step 2.

Iteration 5
Step 2 Present multi-criteria alternative a_5 to cluster $r = 1$,2
$d(\boldsymbol{a}_5, \boldsymbol{w}_1) = 0.246$ and $d(\boldsymbol{a}_5, \boldsymbol{w}_2) = 3.140$, hence a_5 belongs to cluster 1
Set $g_r = 5$.
Step 3 If $s = 5 < p = 6$, set $s = s + 1 = 6$, go to Step 2.

Iteration 6
Step 2 Present multi-criteria alternative a_6 to cluster $r = 1$,2
$d(\boldsymbol{a}_6, \boldsymbol{w}_1) = 2.938$ and $d(\boldsymbol{a}_6, \boldsymbol{w}_2) = 1.61$, hence a_6 belongs to cluster 2
Set $g_r = 6$.
Step 3 If $s = 6 = p = 6$, Stop.

The clustering result for multi-criteria alternatives is:

Cluster number	*Alternatives*	*Cluster center*
1	a_1, a_3, a_5	(1.7,5.1)
2	a_2, a_4, a_6	(4.2,3.25)

Appendix 2: Details of Example 3 for clustering

Example 3
In this example, 2 × 100 alternatives are randomly generated (see Table 2) and the mutli-criteria clustering algortihm is used to solve the problem. The initial values used in the algorithm are set as follows: $k_1 = k_2 = \cdots = k_n = 0.5, \beta = 0.5, \partial = 0.5, \mu = 0.18, p = 100$.
See Table 3 for clusters.

Appendix 3. Examples for selection of the best alternative using Feedforward ANN

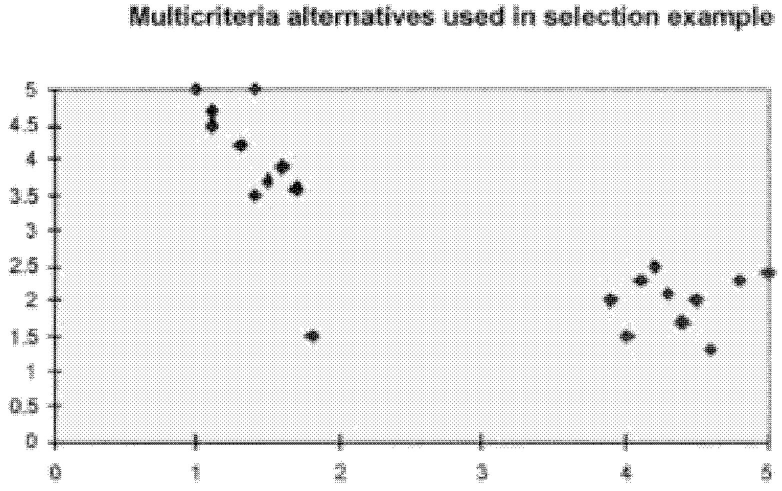
a. *Two attribute example*

Considering the following example—

Cluster characteristics—choosing one or more clusters

The maximum and minimum values of the objective functions, along with cluster mean and variance are presented to the decision-maker.

The two clusters represent different spheres of decision making. Two decision-makers with different preference structure for each cluster will be involved in decision making.



Graph 2. Depicts the 2 dimensional alternatives used in selection example.

Table 6. Presents a table of 20 alternatives, which have 2 attributes

Number	*Alternative*	*Number*	*Alternative*
1	(1, 5)	11	(4, 1.5)
2	(1.1, 4.5)	12	(3.9, 2)
3	(1.1, 4.7)	13	(4.1, 2.3)
4	(1.4, 5)	14	(4.2, 2.5)
5	(1.3, 4.2)	15	(4.3, 2.1)
6	(1.4, 3.5)	16	(4.4, 1.7)
7	(1.5, 3.7)	17	(4.5, 2)
8	(1.6, 3.9)	18	(4.6, 1.3)
9	(1.7, 3.6)	19	(5, 2.4)
10	(1.8, 1.5)	20	(4.8, 2.3)

Table 7. Presents a table of alternatives and the clusters they are grouped under

Cluster number	*Alternatives in the cluster*	*Cluster center*
1	1, 2, 3, 4, 5, 6, 7, 8, 9, 10	(1.39, 3.96)
2	11, 12, 13, 14, 15, 16, 17, 18, 19, 20	(4.38, 2.01)

Table 8. Presents a table of cluster mean and cluster variance and minimum and maximum values with respect to the objective functions on the cluster

Cluster #	*Cluster mean*	*Cluster variance*	f_1		f_2	
			Max	*Min*	*Max*	*Min*
1	(1.39, 3.96)	2.273	1.8	1	5	1.5
2	(4.38, 2.01)	1.612	5	3.9	2.5	1.3

Choosing an alternative for cluster 1

Step 1 Present $(k+1)$ 3 multicriteria alternatives to a decision-maker. Consider that decision maker's unknown utility function is represented by 2 attribute additive exponential function. $U(f) = 0.1f_1^{f2} + 0.1f_1^{f2} + 0.2f_2^2 + 0.2f_1f_2$. The utility values for cluster #2 is listed in Table 9.

Table 9. Presents 3 alternatives and their utility value derived from decision-maker's utility function

Alternative	*Utility value*
(4, 1.5)	(4.05)
(3.9, 2)	5.40
(4.1, 2.3)	7.19

Step 2 Present the above alternatives to a feedforward ANN and train it.

Features of the feedforward neural network—

Number of input nodes—2
Number of output nodes—1
Momentum rate—0.9
Learning rate—0.7
Maximum total error—0.000001
Maximum individual rate—0.000001
Number of hidden layers—1
Number of nodes in the hidden layer—2

Step 3 Present $(k+1)=3$ different multi-criteria alternatives to the trained network.

Step 4 The results from previous step is presented to the decision-maker. Decision-maker agrees with the results obtained from feedforward neural network.

Step 5 Present all the alternatives in cluster 1 to the network and obtain its utility value.

Table 10. Presents the utility values simulated from the trained network and their error values compared to that generated by decision-maker's utility function

Alternative	*Utility value from DM normalized*	*Utility value from ANN normalized*	*Error*
(4.2, 2.5)	0.423	0.398	0.025
(4.3, 2.1)	0.323	0.325	0.002
(4.4, 1.7)	0.254	0.265	0.011

Table 11. Presents the final utility value for all alternatives in cluster 1

Alternative	*Utility value*
(4, 1.5)	0.0379
(3.9, 2)	0.0399
(4.1, 2.3)	0.0420
(4.2, 2.5)	0.0434
(4.3, 2.1)	0.0416
(4.4, 1.7)	0.0400
(4.5, 2)	0.0419
(4.6, 1.3)	0.0388
(5, 2.4)	0.0459
(4.8, 2.3)	0.0446

Step 6 The alternative (4.8, 2.3) has the highest utility value, hence the selected alternative.

Similarly for the cluster 2, assuming decision-maker's preference structure is given by $U(f) = 0.1f_1 + 0.3f_2 + 0.5f_1f_2$. The decision-maker follows the same steps for selection as for cluster 1 above. The utility values for the alternatives in cluster 2 is presented below—

Table 12. Presents the final utility value for all alternatives in cluster 2

Alternative	*Utility value*
(4, 1.5)	0.071
(3.9, 2)	0.091
(4.1, 2.3)	0.105
(4.2, 2.5)	0.115
(4.3, 2.1)	0.095
(4.4, 1.7)	0.079
(4.5, 2.0)	0.092
(4.6, 1.3)	0.065
(5, 2.4)	0.111
(4.8, 2.3)	0.106

The alternative (4.2, 2.5) has the highest utility value, hence the selected alternative from cluster 2.

b. 3 *attribute example*

Cluster characteristics—choosing a cluster

The maximum and minimum values of the objective functions, along with cluster mean and variance are presented to the decision-maker.

Depending upon cluster mean, variance and

Table 13. Presents 3 attribute alternatives used for clustering and selection

#	*Alternative*	#	*Alternative*	#	*Alternative*
1	(0.39, 0.77, 0.16)	10	(0.06, 0.87, 0.67)	19	(0.75, 0.04, 0.33)
2	(0.71, 0.77, 0.83)	11	(0.35, 0.32, 0.29)	20	(0.66, 0.79, 0.91)
3	(0.69, 0.07, 0.46)	12	(0.23, 0.64, 0.04)	21	(0.37, 0.12, 0.86)
4	(0.63, 0.59, 0.55)	13	(0.62, 0.07, 0.52)	22	(0.44, 0.64, 0.83)
5	(0.19, 0.28, 0.36)	14	(0.32, 0.35, 0.38)	23	(0.86, 0.02, 0.18)
6	(0.03, 0.55, 0.07)	15	(0.63, 0.62, 0.62)	24	(0.56, 0.40, 0.24)
7	(0.75, 0.00, 0.23)	16	(0.36, 0.94, 0.51)	25	(0.72, 0.46, 0.19)
8	(1.00, 0.39, 0.79)	17	(0.77, 0.56, 0.35)		
9	(0.85, 0.33, 0.82)	18	(0.39, 0.98, 0.57)		

Table 14. Depicts the results of clustering 25 alternatives into 4 clusters

Cluster #	*Alternative*	*Cluster center*
1	1, 6, 7, 11, 12, 23, 24, 25	(0.49, 0.39, 0.18)
2	2, 8, 9, 20, 21, 22	(0.67, 0.51, 0.84)
3	3, 5, 13, 14, 17, 19	(0.56, 0.23, 0.40)
4	4, 10, 15, 16, 18	(0.41, 0.80, 0.58)

Table 15. Presents a table of cluster mean and cluster variance and minimum and maximum values with respect to the objective functions on the cluster

Cluster #	*Cluster Mean*	*Cluster Variance*	f_1		f_2		f_3	
			Max	*Min*	*Max*	*Min*	*Max*	*Min*
1	(0.49, 0.39, 0.18)	0.0676	0.86	0.03	0.77	0.00	0.29	0.04
2	(0.67, 0.51, 0.84)	0.0581	1.00	0.37	0.79	0.12	0.91	0.79
3	(0.56, 0.23, 0.40)	0.0505	0.77	0.19	0.56	0.04	0.52	0.33
4	(0.41, 0.80, 0.58)	0.0530	0.63	0.06	0.98	0.59	0.67	0.51

maximum and minimum values of the objective function, the decision-maker may choose one or more than one cluster for further consideration. Each cluster will have different preference structure with respect to the same decision-maker.

Choosing an alternative from cluster 2

Since the chosen cluster contains limited number of alternatives, in these cases the selection process does not follow the steps outlined before but decision-

Table 16. Presents 6 alternatives and their utility value derived from decision-maker's utility function

Alternative	*Utility value*
(0.71, 0.77, 0.83)	0.800
(1.00, 0.39, 0.79)	0.596
(0.85, 0.33, 0.82)	0.441
(0.66, 0.79, 0.91)	0.824
(0.37, 0.12, 0.86)	0.211
(0.44, 0.64, 0.83)	0.423

maker would provide utility values for these alternatives. Consider that decision-maker's unknown utility function is represented by 3 attribute additive exponential function. $U(f) = 0.1f_1^3 + 0.2f_2^3 + 0.3f_3^3 + 0.6f_1^2 f_2 + 0.6f_1 f_2^2 + 0.036f_1 f_2 f_3$. The utility values for alternatives in cluster #2 are—

The alternative (0.66, 0.79, 0.91) has the highest utility value, hence the selected alternative.

Similar steps are followed to choose the best alternatives from other clusters.

References

Chan, H. M. and Milner, D. A. (1982) Direct clustering algorithm for group formation in cellular manufacture. *Journal of Manufacturing Systems*, **9**, 65–75.

Day, W. H. E. (1986a) Concensus classification. *Journal of Classification*, **3**, 2.

Day, W. H. E. (1986b) Foreword: Comparison and consensus of classifications. *Journal of Classification*, **3**, 183–185.

Dyer, J. S., Fishburn, P. C. and Steuer, R. E., Multiple criteria decision making, Multiattribute utility theory: The next ten years. *Management Science*, **38**.

Ferligoj, A. and Batagelj, V. (1992) Direct multicriteria clustering algorithms. *Journal of Classification*, **9**(1), 43–61.

Goicoechea, A., Hansen, D. R. and Duckstein, L. (1982) *Multi-objective Decision Analysis with Engineering and Business Applications*, J. Wiley.

Grossberg, S. (1976) Adaptive pattern classification and universal recording. I: parallel development and coding of neural feature detectors. *Biological Cybernetics*, **23**, 121–134.

Grossberg, S. (1987) Competitive learning: from interactive activation to adaptive resonance. *Cognitive Science*, **11**, 23–63.

Hwang, C. L. and K. Yoon, (1981) *Multiple Attribute Decision Making: Methods and Applications*, Springer-Verlag, Berlin; p. 186.

Kamal, S. and Burke, L. I. (1994) FACT: A new neural network-based clustering algorithm for group technology. *International Journal of Production Research*, **34**.

King, J. R. (1980b) Machine-component grouping in production flow analysis: an approach using a rank order clustering algorithm. *International Journal of Production Research*, **18**, 213–232.

Kohonen, T. (1984) *Self-organization and Associative Memory*, Springer-Verlag, Berlin.

Kohonen, T. (1988) An introduction to neural computing. *Neural Networks*, **1**, 3–16.

Kusiak, A. (1987) The generalized group technology concept. *International Journal of Production Research*, **25**, 561–569.

Kusiak, A. and Heragu, S. (1987) Group technology. *Computer in Industry*, **9**, 83–91.

Malakooti, B. and Zhou, Y. (1994) Feedforward artificial neural network for solving discrete multiple criteria decision making problem. *Management Science*, **40**, 1542–1561.

Malakooti, B. and Yang, Z. (1995) Avariable unsupervised learning clustering neural network approach with application to machine-part group formation, *International Journal of Production Research*, **33**(9), 2395–2413.

Malakooti, B. and Zhou, Y. (1998) Approximating polynomial functions by feedforward artificial neural networks. capacity, analysis, and design. *Applied Mathematics and Computation*, **90**, 27–52.

Malakooti, B. and Subramanian, S. (1998) Generalized polynomial decomposable multiple attribute utility functions for ranking and rating of alternatives *AMC*, **91**.

Malakooti, B. and Raman, V. An interactive artificial neural network approach for machine set-up problems. *Journal of Intelligent Manufacturing*, **9** (In press).

Malakooti, B., Zhou, Y., Tandler, E. C. (1995) In-process regressions and adaptive neural networks for monitoring and supervising machining operations. *Journal of Intelligent Manufacturing*, **6**, 53–66.

Pao, Y. H. (1989) *Adaptive Pattern Recognition and Neural Networks*, Addison-Wesley, Reading, MA.

Rumelhart, D. E. and Zipser, D. (1985) Feature discovery by competitive learning. *Cognitive Science*, **9**, 75–112.

Rumelhart, D. E. Hinton, G. and Williams, R. (1986) Learning internal representations by back-propagating errors. *Nature*, **323**, 533–536.

Steuer, R. H. (1986) *Multiple Criteria Optimization: Theory, Computation, and Application*, J. Wiley, New York.

Sun, M. H., Stam, A and Steuer, R. E. Solving multiple objective programming problems using feed-forward artificial neural networks: The Interactive FFANN Procedure, *Manage SCI* **42**(6), 835–849.

Stam, A., Sun, M. H., Haines, M. (1996) Artificial neural network representations for hierarchical preference structures, Comput Oper Res **23**(12), 1191–1201.

Wang, J. and Malakooti, B. (1992) A feedforward neural network for multiple criteria decision making, *Computers and Operations Research*, **19**(2), 151–167.