Control
and Cybernetics
VOL. 15 (1986) No. 2

Clustering analysis: models and algorithms

by

ANDREW KUSIAK

Department of Industrial and Mechanical Engineering
University of Manitoba
Winnipeg, Manitoba R3T 2N2, Canada

ANTHONY VANNELLI

Department of Industrial Engineering
University of Toronto
Toronto, Ontario M5S 1A4, Canada

K. RAVI KUMAR

Department of Business Administration
University of Illinois at Champaign-Urbana
Champaign, Il 61820, U.S.A.

In this paper, the problem of clustering observations into homogeneous groups based on given characteristics of the observations is analyzed. Three distinct integer programming formulations covering important variations of the clustering problem are developed. These variations include finding natural clusters, constraining the number of clusters and restricting the size of clusters. Efficient heuristic techniques employing Lagrangian and eigenvector based methods are developed to solve these problems.

1. Introduction

Classification has a rich history, but numerical methods used for the purpose of classification are fairly recent. The major developments have occurred in the last two decades. Sokal and Sneath (1963) published one of the first books on this subject.

Classification (or typology) is concerned with the identification of an observation and its placement into a homogeneous group based on some characteristics. The pursuit of classification can be seen in all fields. For example, in judical science, the Supreme Court judges may be grouped on the basis of their legal opinions on a sample of cases. In psychology and consumer behaviour, people may be classified according to their personality and taste characteristics. In international marketing,

the world markets can be classified into segments based on cultural, socio-economical and political characteristics. In strategic management, firms are classifed according to the production, financial and marketing strategies used. In engineering design, parts produced are classified according to the geometrical, tolerance and machining characteristics they possess. Classification and cluster analysis has been applied in the following areas:

> biology (Everitt, 1980), data reorganization (McCormick et al., 1972), medicine (Klastorin, 1982), pattern recognition (Tou and Gonzalez, 1974), part selection in automated systems (Kusiak, 1985a), production flow analysis (King, 1980), race mixture study (Rao, 1977), task selection (Nagai et al., 1980), control engineering (Siljak, 1984).

In the cases, where it is possible to specify groups a priori, statistical techniques such as multiple discriminant analysis provide an analytical method to define topology functions (Green, 1978). But when it is not possible to specify these groups, one needs to resort to various combinatorial algorithms and heuristics to aid in constructing the clusters.

An assumption underlying the use of clustering techniques is that homogeneous clusters actually exist in the data. The basic problem in cluster analysis is to devise algorithms and heuristics that group entities into clusters based on observed attributes. The development of these heuristics and algorithms have typically depended on conceptual representation of the process of clustering. These representations have been largely visual and can be of two distinct types, matrix representations and graph representations.

Matrix representations have usually been used in the domain of social sciences. One of the first applications in marketing segmentation and selection was by Green et al. (1967) who desired to match representative test market with larger product markets. Here, a variety of market characteristics were gathered for a number of potential test markets and arranged in a matrix-type representation with rows representing cities and columns, the market characteristics. The object was to rearrange all those rows, which were "similar", such that they were adjacent in the permuted matrix. As is often the case, the market characteristics were measured in different scales, and therefore, had to be normalised (re-scaled to have a mean of 0 and a standard deviation of 1) before similarity measurements using weighted Euclidean distances were used.

Another application of the matrix representation is in the area of group technology, which concerns itself with grouping machines (and consequently, the parts that can be produced on the machines) so as to form independent manufacturing cells (Burbidge, 1975 and King, 1980). In this application, rows represent the machines and columns represent the parts produced. The matrix entries are binary, 1 representing the use of the machine for the part and 0 otherwise. The object is to permute the rows and columns so as to obtain a block diagonal representation of the original matrix, with each block representing a cluster.

Graph representations have usually been used in the engineering sciences field, particularly electrical engineering. One application arises in the design and monitoring of power system operations (Stagg et al., 1970 and Bills, 1970). Here, a weighted graph representation is used to depict the network of power grid buses, with the nodes representing the machines, such as transformers, and the edges (or arcs) representing the interlinking connections between these buses. The admittance between these buses is taken as the weight on (or capacity of) the edges. The object is to decompose the graph into sub-graphs (by deleting edges) such that there are minimal interconnections between the sub-graphs (and hence maximal connections within the sub-graphs).

Another application arises in the design of very large scale integration (VLSI) circuits (Kernighan and Lin, 1970). The circuits are represented as graphs with the electrical elements, such as resistors, being the nodes of the graph and the wiring between these elements representing the edges. The purpose of this representation is to find a way to partition this graph so as to maximize the number of circuits that can be packed into the chips.

In this paper, we describe three distinct integer programming formulations which cover important variations of these two representations. We characterize the integer programming formulations by two constraints:

(1) fixed number of clusters

(2) restriction on the number of elements within each cluster.

The three integer programming formulations presented allow one to deal with these two constraints. The first formulation (P1) does not incorporate any of these constraints: that is, we allow the algorithm to generate natural clusters. Since many clusters could be generated by the first formulation, a second formulation (P2) is developed which restricts the number of clusters. Finally, we consider a model which allows one to deal with restrictions on the number of clusters and cluster size.

The paper is divided into five sections. In Section 2, we discuss a clustering problem with no restrictions on the number of clusters and cluster sizes. A clustering problem with a fixed number of clusters is presented in Section 3. A Lagrangian relaxation approach is used to solve this problem. In Section 4, we formulate and solve a clustering problem with a fixed number of clusters and cluster sizes. An eigenvector based approach is used in the subsequent analysis. Conclusions are presented in Section 5.

2. A clustering problem without any constraints

2.1. Problem Formulation

Typically, one formulates a clustering problem, where there is no a priori information regarding the number of clusters and cluster sizes. In this case the resulting clusters are usually generated by visual inspection.

Before formulating a clustering problem that does not restrict the number of clusters and cluster sizes, let us consider a 0–1 matrix $A=[a_{ij}]_{m\times n}$. For any two row vectors $a_i=[a_{i1}, ..., a_{ik}, ..., a_{in}]$ and $a_j=[a_{j1}, ..., a_{jk}, ..., a_{jn}]$ of matrix A, define a distance

$$d_{ij}=\sum_{k=1}^{n}\delta(a_{ik}, a_{jk}) \tag{1a}$$

where

$$\delta(a_{ik}, a_{jk})=\begin{cases}1 & \text{if } a_{ik}=a_{jk}=1\\ 0 & \text{otherwise}\end{cases} \tag{1b}$$

In this clustering problem, we attempt to permute rows and columns of matrix A to maximize the sum of the distances d_{ij} (d'_{ij}) between any two adjacent rows (columns), respectively. It can be formulated as follows:

(P1)
$$\max D=\sum_{i=1}^{m-1}\sum_{j=i+1}^{m} d_{ij}+\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d'_{ij} \tag{2}$$

for all $n!\,m!$ possible matrices obtained permuting rows and columns of the initial matrix A.

Lenstra (1974) has shown that problem (P1) is equivalent to two travelling salesman problems. Based on this fact the following two conclusions can be drawn:
(1) this clustering problem is an NP-complete problem
(2) a travelling salesman algorithm can be applied to solve the clustering problem.

2.2. Algorithms for solving problem (P1)

To date a large number of algorithms for solving problem (P1) have been developed by researchers working in many different areas. Some of the most efficient heuristic algorithms have been discussed in Kusiak (1985), namely:

(1) McCormick et al. (1972)
(2) Bhat and Haupt (1976)
(3) King (1980, 1982)
(4) rank energy (Kusiak, 1985).

All of these algorithms are based on rearranging rows and columns of matrix A to produce some visible clusters. The difference between them is in the way this rearrangement is performed.

Computational complexity of each of these algorithms is shown in Table 1.

Table 1. Computational Complexities of Clustering Algorithms

McCormick et al (1972)	Bhat and Haupt (1976)	King (1982)	Rank Energy
$O_M(nm^2+n^2m)$,	$O_B(nm^2+n^2m)$	$O_K(nm^2+n^2m)$	$O_R(nm^2+n^2m)$

3. A clustering problem with fixed number of clusters

3.1. Problem Formulation

In order to formulate this problem let us introduce the following notation:

n number of elements

m required number of clusters

d_{ij} distance from element i to element j ($d_{ij} \geqslant 0$), $\forall i \neq j = 1, \ldots, n$ and $d_{ij} \neq 0$, $\forall i = j = 1, \ldots, n$).

$$x_{ij} = \begin{cases} 1 \text{ if } i^{\text{th}} \text{ element belongs to } j^{\text{th}} \text{ cluster} \\ 0 \text{ otherwise} \end{cases}$$

The objective function minimizes the total sum of distances:

$$Z_x = \min Z(x) = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} x_{ij} \tag{3}$$

$$\text{s.t.} \sum_{j=1}^{n} x_{jj} = 1 \quad \forall i = 1, \ldots, n \tag{4}$$

(P2)

$$\sum_{j=1}^{n} x_{ij} = m \tag{5}$$

$$x_{ij} \leqslant x_{jj} \qquad \forall i = 1, \ldots, n \qquad \forall j = 1, \ldots, n \tag{6}$$

$$x_{ij} = 0,1 \qquad \forall i = 1, \ldots, n \qquad \forall j = 1, \ldots, n \tag{7}$$

Constraint (4) ensures that each element belongs to exactly one cluster. Constraint (5) specifies a required number of clusters. Constraint (6) ensures that element j belongs to cluster j only when this cluster is formed. The last constraint (7) imposes integrality.

3.2. A Subgradient Algorithm

Problem (P2) has been solved by Mulvey and Crowder (1979) but a more efficient subgradient algorithm is presented here. The main difference between the proposed algorithm and that of Mulvey and Crowder (1979) is in the procedure of computing lower bounds. The algorithm of Mulvey and Crowder (1979) computes the lower bounds based upon a heuristic algorithm developed by Ward (1963). The presented subgradient algorithm is based on a simple procedure of computing lower bounds shown in Arthanari and Dodge (1981).

Dualizing on constraint (4) the objective function (3) is transformed as follows (for $u_i \geq 0$, $\forall i=1, ..., n$)

$$Z_u=\min Z(u_i)=\sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}\, x_{ij}+\sum_{i=1}^{n} u_i\left(1-\sum_{j=1}^{n} x_{ij}\right) \tag{8}$$

Reordering (8) the following relaxed problem is obtained

$$Z_u=\min Z(u_i)=\sum_{i=1}^{n}\sum_{j=1}^{n}(d_{ij}-u_i)\, x_{ij}+\sum_{i=1}^{n} u_i \tag{9}$$

(P_u)

s.t. (5), (6) and (7).

The best choice of u is an optimal solution to the dual problem

$$Z_D=\max_u Z_u \tag{10}$$

(D)

s.t. (5), (6) and (7).

Framework of the Subgradient Algorithm

In the subgradient algorithm one specifies initial values of Lagrangian multipliers u_i^0 and in each iteration $k+1$ an updated sequence u^{k+1} is generated as follows:

$$u^{k+1}=u_i^k+\alpha^k\, g_i^k \tag{11}$$

where: α^k is a positive scalar step size
g_i^k is a subgradient; in the case of the problem (P2)

$$g_i=1-\sum_{j}^{n} x_{ji}^*, \tag{12}$$

where x_{ji}^* is an optimal solution to the problem (P_u)

The most commonly used step size is

$$\alpha^k=\frac{\gamma^k\,(UB^k-Z_u^k)}{\|g^k\|}, \tag{13}$$

where: γ^k is a scalar statisfying $0<\gamma^k<2$ (see Motzkin, 1954)
UB^k is an upper bound on Z_D
$\|\bullet\|$ is an Euclidean norm.

To compute UB^k in our subgradient algorithm a simple heuristic, generating a feasible solution to the problem (P2) is used.

In order to solve the dual problem (D) the following general framework of a subgradient algorithm is applied:

Step 0. Set iteration number $k=0$ and choose initial values of Lagrangian multipliers u_i^k, $i=1, ..., n$.

Step 1. Solve problem (P_u) for all u_i^k. The value obtained Z_u is a lower bound on the value of the objective function Z_D in (D).

Step 2. Generate a feasible solution to problem (P2). The value Z_x is an upper bound on the value of the objective function Z_D in (D).

Step 3. If the current solution to the problem (D) satisfies a given stopping criterion, stop; otherwise go to Step 1.

LOWER BOUNDS PROCEDURE

A procedure for computing the lower bounds given in Arthanari and Dodge (1981) will be applied. Denote:

$$s_{ij}=\min\{d_{ij}-u_i, 0\} \tag{14}$$

and let $S_j=\sum_{i=1}^{n} s_{ij}$.

To minimize (9) let us arrange the first m values of S_j in an increasing order $S_{j(1)}\leqslant S_{j(2)}\leqslant \ldots \leqslant S_{j(m)}$ and let the set $\{j(1), j(2), ..., j(m)\}=L$.

The optimal solution to the problem (P_u) is then

$$x_{ij}^*=\begin{cases}1 & \text{if } i=j\in L\\ 0 & \text{otherwie}\end{cases} \tag{15}$$

and

$$x_{ij}^*=\begin{cases}1 & \text{if } i\neq j\in L\\ 0 & \text{otherwise}\end{cases} \tag{16}$$

Substituting x_{ij}^* of (15) and (16) into (9) a lower bound for the problem (D) is obtained.

UPPER BOUNDS PROCEDURE

A feasible solution to the problem (P2) can be computed in the way shown in Arthanari and Dodge (1981), namely:

$$x_{ij}=\begin{cases}1 & \text{if } i=j\in L\\ 0 & \text{otherwise}\end{cases} \tag{17}$$

and

$$x_{ij}=\begin{cases}1 & \text{if } i\neq j \quad \text{and} \quad d_{ij}=\min\limits_{r\in L} d_{ir}\\ 0 & \text{otherwise}\end{cases} \tag{18}$$

One can easily see that the above solutions satisfy all constraints of problem (P2). Substituting all x_{ij} to (1) an upper bound to the problem (D) is obtained.

SUBGRADIENT ALGORITHM

The algorithm for solving the problem (D) is as follows:

Step 0. Set $k=1$, $u_i^k \geq 0$, $\varepsilon_1>0$, $\varepsilon_2>0$, $\gamma^0>0$, $UB^0=+\infty$, $LB^0=-\infty$, where:

u_i^0 initial value of the Lagrangian multipliers
$\varepsilon_1, \varepsilon_2$, precision values
γ^0 initial value on the scalar $(0<\gamma^0<2)$
UB^0 initial upper bound on (10)
LB^0 initial lower bound on (10)

Step 1. Compute a feasible solution for (P2) from (17) and (18) in order to obtain a value Z_x^k of (3).
Compute an upper bound on (9)

$$UB^k=\min\{UB^{k-1}, Z^k\}.$$

Step 2. Compute the values of x_{ij}^* from (15) and (16) and substitute into (9) to obtain a value Z_u^k for updated values of u_i^k, $i=1, \ldots, n$.
Compute a lower bound on (Z_D)

$$LB^k=\max\{LB^{k-1}, Z_u^k\}.$$

If $Z_u^k<LB^k$, then reduce γ^k.
If $\gamma^k<\varepsilon_1$, stop; otherwise continue.
If $(UB^k-LB^k)/UB^k<\varepsilon_2$, stop; otherwise go to step 3.

Step 3. Compute the following:

(a) subgradients g_i^k at x_{ij}^*

$$g_i^k=1-\sum_{j=1}^{n} x_{ij}^*$$

(b) step size

$$\alpha^k=\frac{-\gamma^k (UB^k-Z_u^k)}{\|g_i^k\|}$$

(c) updated values of Lagrangian multipliers

$$u_i^{k+1}=u_i^k+\frac{\alpha^k g_i^k}{\|g_i^k\|}$$

Set $k=k+1$ and go to Step 1.

3.3. Computational Results

The subgradient algorithm described has been applied to solve a number of problems. For each problem the distances d_{ij} were generated by a uniform, continuous random number generator. Different values of initial parameters u_i^0 and γ^0 have been tested. The algorithm performed well for $u_i^0=1.1 \max_j\{d_{ij}\}$ and $\gamma^0=0.75$ which were determined experimentally.

Tables 2 and 3 show the number of iterations and CPU time (in seconds) for 20 different problems with the precision value $\varepsilon_1=5\%$ and $\varepsilon_2=0.1\%$ respectively.

Table 2. CPU time and number of iterations for problems solved with the precision value $\varepsilon_2=\frac{UB-LB}{UB}100\%\leqslant 5\%$

m \ n	10	20	30	40	50	60	70	80	90	100
5	5	5	4	5	5	5	5	5	5	5
	0.39	1.04	2.05	3.31	4.88	6.65	8,86	11.46	14.40	17.11
10	5	5	4	4	4	4	4	4	4	4
	0.23	0.77	1.26	2.15	3.30	4.65	6.31	8.22	10.28	12.52

Table 3. CPU time and number of iterations for problems solved with the precision value $\varepsilon_2=\frac{UB-LB}{UB}100\%\leqslant 0.1\%$

m \ n	10	20	30	40	50	60	70	80	90	100
5	8	8	8	8	8	8	8	8	8	13
	0.52	1.69	3.28	5.34	7.94	10.87	14.36	18.66	23.17	45.51
10	8	7	7	7	7	7	7	7	7	7
	0.38	1.08	2.28	3.90	5.92	8.34	10.81	14.68	18.50	22.28

As one can see in Tables 2 and 3 the algorithm requires a small number of iterations to generate a good quality feasible solution or in many cases the optimal solution.

To show the efficiency of this algorithm we have solved five sets of different problems by this algorithm and compared results obtained with ones presented by Mulvey and Crowder (1979). Table 4 illustrates this comparison.

Table 4. Comparison of the Subradient Algorithm to the Algorithm of Mulvey and Crowder (1979)

Problem Number	Number of Attributes n	Number of Iterations for $m=5$	
		Mulvey and Crowder (1979) Algorithm	Proposed Algorithm
1	25	26	6
2	50	74	7
3	70	22	7
4	80	82	7
5	100	25	7

Al the above computations were performed on a CDC CYBER 170–720 computer. The algorithm presented requires on average much smaller number of iterations than the Mulvey and Crowder (1979) algorithm to solve a problem of the same size.

4. A clustering problem with fixed number of clusters and cluster sizes

4.1. Problem Formulation

The clustering problem formulations described in the last two sections may not necessarily generate desirably sized clusters. Very large clusters or a large number of very small clusters may be a consequence of these clustering algorithms. In this section, we formulate an eigenvector based approach which allows a fixed number of clusters of fixed size to be generated. We begin by introducing the following two definitions. Consider an undirected graph $G=(V, E)$ where d_{ij} is a distance measure between elements v_i and v_j.

DEFINITION 1. A k-cluster of $G(V, E)$ is obtained by deleting the edges of G to obtain k disconnected subgraphs $G_i=(V_i, E_i)$, $i=1, 2, ..., k$ and $\bigcup_{i=1}^{k} V_i=V$.

DEFINITION 2. An optimal k-cluster is a k-cluster which maximizes the sum of the intra-cluster distance of the k clusters.

The optimal k-clustering problem is a generalization of the k-means problem (Hartigan, 1975). The main difference is that we impose a limit on the cluster size. We formalate the optimal k-clustering problem as a 0–1 quadratic programming (0–1 QP) problem. Since n elements are to be divided among k clusters, we assign to each element i the variable $x_{i1}, x_{i2}, ..., x_{ik}$ where

$$x_{ij}=\begin{cases}1 & \text{if element } i \text{ is assigned to cluster } j\\ 0 & \text{otherwise}\end{cases}$$

Each element i is in exactly one cluster, thus

$$\sum_{j=1}^{k} x_{ij}=1, \quad \forall i=1, 2, ..., n \tag{19}$$

Cluster j has exactly m_j elements in it. Therefore, we add the following set of constrains to (19)

$$\sum_{i=1}^{n} x_{ij}=m_j, \quad \forall j=1, 2, ..., k$$

Since each edge in cluster l ($l=1, 2, ..., k$) is represented by the node product $x_{il}\, x_{jl}$, then the edge joining element i to element j is included in cluster I if and only if $x_{il}=x_{jl}=1$.

If d_{ij} is the distance between elements i and j, then the total distance of all distances in all k clusters is given by

$$\sum_{l=1}^{k} \sum_{i=1}^{n-k} \sum_{j=i+1}^{n} d_{ij} x_{il} x_{jl}$$

The 0–1 QP formulation of the optimal k-clustering problem is:

(P3)

$$\min \sum_{l=1}^{k} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d_{ij} x_{il} x_{jl} \tag{20}$$

$$\text{s.t.} \sum_{j=1}^{k} x_{ij}=1, \quad \forall i=1, 2, ..., n \tag{21}$$

$$\sum_{i=1}^{k} x_{ij}=m_j, \quad \forall j=1, 2, ..., k \tag{22}$$

$$x_{ij}=0 \quad \text{or} \quad 1, \quad \forall i=1, 2, ..., n; \ \forall j=1, 2, ..., k \tag{23}$$

4.2. An Approximation Algorithm for Solving Problem (P3)

An eigenvector based approach is described for finding an approximate solution to problem (P3). This eigenanalysis approach is a simple extension of an approach used by Barnes (1982) and Vannelli (1984) to partition the nodes of a graph subject to the constraints given in (P3). In this case one is maximizing the objective function in (P3). Clearly, problem (P3) is equivalent to

(NP3)

$$\max \sum_{l=1}^{k} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (-d_{ij}) x_{il} x_{jl} \tag{24}$$

$$\text{s.t.} \sum_{j=1}^{k} x_{ij}=1, \quad \forall i=1, ..., n \tag{25}$$

$$\sum_{i=1}^{n} x_{ij}=m_j, \quad \forall j=1, ..., k \tag{26}$$

$$x_{ij}=0 \quad \text{or} \quad 1, \quad \forall i=1, ..., n; \ \forall j=1, ..., k. \tag{27}$$

Given that $-d_{ij} \in -D$, Barnes (1982) shows that problem (NP3) can be approximated by the linear transportation problem

(TP3)

$$\max \sum_{j=1}^{k} \sum_{i=1}^{n} \frac{u_{ij}}{\sqrt{m_i}} x_{ij} \tag{28}$$

$$\text{s.t.} \sum_{j=1}^{k} x_{ij}=1, \quad \forall i=1, ..., n \tag{29}$$

(TP3)

$$\sum_{i=1}^{n} x_{ij}=m_j, \quad \forall j=1, \ldots, k \tag{30}$$

$$x_{ij}\geqslant 0, \quad \forall i=1, \ldots, n \quad \forall j=1, \ldots, k \tag{31}$$

where $\lambda_1\geqslant \ldots \geqslant\lambda_k$ are the k largest eigenvalues of $-D$ (k smallest eigenvalues of D) and $u_1, u_2, \ldots, u_k$ are the corresponding eigenvectors.

The linear transportation problem can be solved in the worst case in $0(n^3)$ time (Lawler, 1976).

4.3. A Numerical Example

We apply the approximation algorithm given in Section 4.2 on the following food data problem given in Hartigan (1975, pp. 88)

Table 5. Clustering Problem from Hartigan (1975)

Food Type	Energy	Protein	Calcium
1	13	21	1
2	5	36	1
3	5	37	2
4	11	29	1
5	8	30	1
6	12	27	1
7	6	31	2
8	4	29	1

Consider the distance measure

$$d_{ij}=\|a_i-a_j\|^2$$

where a_i is the i^{th} food type row. For example,

$$d_{12}=(13-5)^2+(21-36)^2+(1-1)^2=289.$$

The 8×8 distance matrix D of the food data representation of Table 5 is

$$D=\begin{bmatrix} 0 & 289 & 321 & 68 & 106 & 37 & 150 & 145 \\ 289 & 0 & 2 & 85 & 45 & 130 & 27 & 50 \\ 321 & 2 & 0 & 101 & 59 & 150 & 37 & 66 \\ 68 & 85 & 101 & 0 & 10 & 5 & 30 & 49 \\ 106 & 45 & 59 & 10 & 0 & 25 & 6 & 17 \\ 37 & 130 & 150 & 5 & 25 & 0 & 53 & 68 \\ 150 & 27 & 37 & 30 & 6 & 53 & 0 & 9 \\ 145 & 50 & 66 & 49 & 17 & 68 & 9 & 0 \end{bmatrix} \tag{32}$$

If we wish to find two optimal clusters of D where each group has four elements, we find the two largest eigenvalus of $-D$ which are 488.4 and 98.059 respectively. The corresponding eigenvectors are

$$u_1^T=[.655,\ -.447,\ -.495,\ .1102,\ -.0503,\ .259,\ -.1709,\ -.124]^T$$
$$u_2^T=[.363,\ .1658,\ .338,\ -.405,\ -.4737,\ .308,\ -.3798,\ -.315]^T$$

The transportation problem approximation of the optimal k-cluster problem (P3) is

$$\text{Max } 1/2 \sum_{j=1}^{2} \sum_{i=1}^{8} u_{ij}\, x_{ij}$$
$$\text{s.t. } x_{11}+x_{12}=1$$
$$x_{21}+x_{22}=1$$
$$\vdots$$
$$x_{81}+x_{82}=1$$
$$x_{11}+x_{21}+\ \dots\ +x_{81}=4$$
$$x_{ij}\geqslant 0$$

The solution of this problem is to group elements 1, 4, 5, and 6 in one cluster and the others in the second cluster. The resulting clusters are obtained by permuting the rows and columns of D into

$$\bar{D}=\left[\begin{array}{cccc:cccc} 0 & 10 & 68 & 5 & 30 & 49 & 85 & 101 \\ 10 & 0 & 106 & 25 & 6 & 17 & 45 & 59 \\ 68 & 106 & 0 & 37 & 150 & 145 & 289 & 321 \\ 5 & 25 & 37 & 0 & 53 & 68 & 130 & 150 \\ \hdashline 30 & 6 & 150 & 53 & 0 & 9 & 27 & 37 \\ 49 & 17 & 145 & 68 & 9 & 0 & 50 & 66 \\ 85 & 45 & 289 & 130 & 27 & 50 & 0 & 2 \\ 101 & 59 & 321 & 150 & 37 & 66 & 2 & 0 \end{array}\right]$$

Note that the sum of the intra-cluster elements in $\bar{D}$ is small in this case.

5. Conclusions

The clustering problem has been of interest to many researchers working in different areas. In this paper, an attempt has been made to present a uniform view of the clustering problems. Two popular representations of this problem are matrix

models and graph models. In the matrix representation, rows are rearranged such that "similar" rows are adjacent in the permuted matrix. In clustering problems modelled by graphs, the object is to decompose the graph into sub-graphs such that there are minimal interconnections between the sub-graphs.

Three distinct integer programming formulations, which cover important variations of these two representations were presented. First, we considered the problem of finding natural clusters. The problem was shown to be equivalent to two travelling salesman problems, which can be solved by efficient heuristic techniques. Second, a clustering problem with a fixed number of clusters was discussed. A Lagrangian relaxation method was developed for solving this problem. An efficient subgradient algorithm was developed and was shown to require a much smaller number of iterations than the Mulvey and Crowder (1979) algorithm. Finally, a clustering problem with a fixed number of clusters and cluster sizes was formulated. An eigenvector approach led to an approximation of the original problem by a linear transportation problem.

6. Acknowledgements

Research of the first author (A. Kusiak) has been partially supported by the Natural Sciences and Engineering Research Council of Canada. Research of the second author (A. Vannelli) has been partially supported by the Natural Sciences and Engineering Research Council of Canada (Grant No. U0381).

References

[1] ARTHANARI I. S., DODGE Y. Mathematical Programming in Statistics. New York, Wiley, 1981.

[2] BARNES E. R. An algorithm for partitioning the nodes of a graph. *SIAM J. of Algebraic and Discrete Methods.* **3** (1982), 541–550.

[3] BILLS G. W. On-line Stability Analysis Study, Final Report for Project RP90-1, Edison Electric Institute, 1970.

[4] BURBIDGE J. L. The Introduction of Group Technology New York, Halsted Press, John Wiley, 1975.

[5] BHAT M. V., HAUPT A. An efficient clustering algorithm. *IEEE Transactions on Systems Man and Cybernetics.* **SMC-6** (1979), 61–64.

[6] EVERITT B. Cluster Analysis. New York, Halsted Press. 1980.

[7] GREEN P. E. Analyzing Multivariate Data. Dryden Press, Hinsdale, 1978, pp. 290–335.

[8] GREEN P. E., FRANK R. E., ROBINSON J. Cluster analysis in test market selection. *Management Science.* **13** (1967) 387–400.

[9] KERNIGHAN B. W., LIN S. An efficient procedure for partitioning graphs, *Bell Systems Technical Journal.* 1970, pp. 291–307.

[10] KING J. R. Machine-component group formation in production flow analysis: An approach using a rank order clustering algorithm, *International Journal of Production Research.* **18** (1980), 213–232.

[11] KLASTORIN T. D. An alternative method for hospital partition determination using hierarchical cluster analysis. *Operations Research.* **30** (1982), 1134–1146.

[12] KUSIAK A. Computer aided data base design. Working Paper No. 9/83, Dept. of Industrial Engineering, Technical University of Nova Scotia, Halifax, N. S., 1983.

[13] KUSIAK A. The part families problem in flexible manufacturing systems. *Annals of Op. Res.*, **3** (1985), 279–300.

[14] KUSIAK A. Flexible manufacturing systems: A structural approach. *International Journal of Production Research*, **23** (1985a), 1057–1073.

[15] LAWLER E. L. Combinatorial Optimization: Networks and Matroids, New York, Holt, Rinehart and Winston, 1976.

[16] LENSTRA J. K. Clustering a data array and the traveling salesman problem, *Operations Research.* **22** (1974), 413–414.

[17] MCCORMICK W. T., SCHWEITZER P. J., WHITE T. W. Problem decomposition and data reorganization by clustering technique, *Operations Research.* **20** (1972), 993–1009.

[18] MOTZKIN T., SCHOENBERG I. J. The relaxation methods for linear inequalities, *Canadian Journal of Mathematics*, **6** (1954), 393–404.

[19] MULVEY J. M., CROWDER H. P. Cluster analysis: An application of Lagrangean relaxation. Management Science. **25** (1979), 329–340.

[20] NAGORI Y., TENDA S., SHINGA T. Determination of similar task types by the use of the multidimensional classification method: towards improving quality of working life and job satisfaction. *International Journal of Production Research.* **18** (1980), 307–322.

[21] RAO G. R. Cluster analysis applied to a study of race mixture in human populations, in: J. V. Ryzin, Ed., Classification and Clustering. New York, Academic Press, 1977.

[22] SILJAK D. D., SEZER M. F. Nested decomposition into weakly coupled components. IFAC 9th World Congress, Budapest, Hungary July 2–6, 1984.

[23] SOKAL R. R., SNEATH P. H. Principles of Taxonomy. London, Freeman, 1963.

[24] STAGG C. W., DOPAZO J. P., KILTON O. A., VAN SLYK L. S. Techniques for real-time monitoring of power operating systems, *IEEE Transactions on PAS* **89** (1970), 545–555.

[25] TOU J. T., GONZALEZ R. C. Pattern Recognition Principles. Reading, Massachusetts, Addison-Wesley, 1974.

[26] VANNELLI A. Approximating a class of graph decomposition problems by linear transportation problems, IBM Research Report RC 10584 (47380), *Journal of Classification* 1985 (to appear).

[27] WARD J. H., Jr. Hierarchical grouping to optimize an objective function. *Journal of American Statistical Association* **58** (1963) 236–244.

Analiza skupień: modele i algorytmy

W artykule analizuje się zagadnienie grupowania wyników obserwacji w jednolite grupy na podstawie danych charakterystyk wyników obserwacji. Podano trzy różne sformułowania zadania programowania całkowitoliczbowego, odzwierciedlające ważne warianty zagadnienia analizy skupień. Warianty te obejmują: znajdowanie naturalnych skupień, ograniczenie ilości skupień oraz ograniczenie liczności skupień. Przedstawiono sprawne techniki heurystyczne rozwiązywania podanych zagadnień, posługujące się mnożnikami Lagrange'a i wartościami własnymi.

Кластерный анализ: модели и алгоритмы

В статьи рассмотривается задача группировки наблюдений в однородные группы, на основе данных характеристик этих наблюдений. Даны три разные формулировки задачи целочисленного программирования отображающей существенные варианты задач кластерного анализа. Эти варианты вмещают в себе: нахождение естественных кластеров, ограничение числа кластеров и ограничение численности кластеров. Представлены эффективные эвристические методы решения приведенных задач. Эти подходы используют множители Лагранжа и собственные значения.