Acknowledgement

The authors wish to thank Ellis Johnson for helpful discussions during the course of this research.

References

[1] J. Aráoz, W.H. Cunningham, J. Edmonds and J. Green-Krótki, "Reductions to 1-matching polyhedra," *Networks* 13 (1983) 455-473.
[2] G. Calvillo, "The concavity and intersection properties for integral polyhedra," in: M. Deza and I.G. Rosenberg, eds., *Annals of Discrete Mathematics* 8 (North-Holland, Amsterdam, 1980) pp. 221-228.
[3] J.O. Cerdeira and C.J. Luz, "Covering nodes with a fixed number of trees," Technical report, Nota no. 3/85, Centro de Estatística e Aplicacões, Lisboa/I.N.I.C.
[4] J. Edmonds, "Matroids and the greedy algorithm," *Mathematical Programming* 1 (1971) 127-136.
[5] J. Edmonds and E.L. Johnson, "Matching: A well-solved class of integer linear programs," in: R.K. Guy, ed., *Combinatorial Structures and their Applications* (Gordon and Breach, New York, 1970) pp. 89-92.
[6] K.G. Murty and C. Perin, "A 1-matching blossom-type algorithm for edge covering problems," *Networks* 12 (1982) 379-391.
[7] W.R. Pulleyblank, "Polyhedral combinatorics," in: A. Bachem, M. Grötschel and B. Korte, eds., *Mathematical Programming-The State of the Art* (Springer-Verlag, Heidelberg, 1983) pp. 312-345.
[8] L.J. White, "Minimum covers of fixed cardinality in weighted graphs," *SIAM Journal of Applied Mathematics* 21 (1971) 104-113.
[9] L.J. White and M.L. Gillenson, "An efficient algorithm for minimum k-covers in weighted graphs," *Mathematical Programming* 8 (1975) 20-42.

Mathematical Programming 45 (1989) 59-96
North-Holland

A CUTTING PLANE ALGORITHM FOR A CLUSTERING PROBLEM

M. GRÖTSCHEL
Institut für Mathematik, Universität Augsburg, Memminger Str. 6, 8900 Augsburg, West Germany

Y. WAKABAYASHI
Universidade de São Paulo, Instituto de Mathemática e Estatística, Caixa Postal 20.570 (Agência Iguatemi), 01498 São Paulo SP, Brazil

Received 12 November 1987
Revised manuscript received 12 April 1988

In this paper we consider a clustering problem that arises in qualitative data analysis. This problem can be transformed to a combinatorial optimization problem, the clique partitioning problem. We have studied the latter problem from a polyhedral point of view and determined large classes of facets of the associated polytope. These theoretical results are utilized in this paper. We describe a cutting plane algorithm that is based on the simplex method and uses exact and heuristic separation routines for some of the classes of facets mentioned before. We discuss some details of the implementation of our code and present our computational results. We mention applications from, e.g., zoology, economics, and the political sciences.

Introduction

The need of analysing data that arise from the measurement of a number of characteristics (or attributes) associated with each object of a given set, occurs very frequently in sociology, zoology, economics, and many other sciences. The areas of study concerned with this type of problem are known as *data analysis*, *multivariate analysis*, and *taxonomy*.

We consider here a problem occurring in qualitative data analysis, of the following type.

Given a data set consisting of the description of a set of objects with respect to a number of characteristics, find a best partition of the object set into "homogeneous" disjoint classes (or clusters).

In this paper we give a precise formulation of this *clustering problem* and show how it can be reduced to a graph optimization problem which we call *clique partitioning problem* (CPP). This is done in Section 2. In Section 3 we summarize some results of Grötschel and Wakabayashi (1987) on the polyhedron associated with the CPP. In Section 4 we describe a cutting plane algorithm for CPP which is based on these theoretical results. Finally, in Section 5 we report on the computational results with our code. Many applications from zoology, marketing, and the

political sciences are given and the optimization process for each of these applications is illustrated.

1. Definitions and notation

We assume that the reader is familiar with the basic concepts of graph theory. The definitions not given here can be found in Bondy and Murty (1976). All graphs we consider are simple. We denote a graph G with node set V and edge set E by $G=(V, E)$. An edge e with endnodes u and v is denoted by uv. If S is a node set of $G=(V, E)$ then we denote the set of edges in G with both endnodes in S by $E(S)$, that is,

$$E(S)=\{uv\in E \mid u, v\in S\}.$$

Moreover, if $S_1, \ldots, S_k$ are subsets of V then

$$E(S_1, \ldots, S_k) := \bigcup_{i=1}^{k} E(S_i).$$

If $S, T\subseteq V$ and $S\cap T=\emptyset$ then

$$[S:T] := \{uv \mid u\in S, v\in T\}$$

denotes the set of edges with one endnode in S and the other in T.

A graph is called *complete* if every pair of its nodes is linked by an edge. A *clique* is a subgraph of a graph that is complete (a clique is not necessarily a maximal complete subgraph). We will denote the (up to isomorphism unique) complete graph with n nodes by $K_n=(V_n, E_n)$.

We say that $\Gamma=\{W_1, \ldots, W_k\}$ is a *partition* of V if $W_i\cap W_j=\emptyset$ for $1\leq i<j\leq k$, $V=W_1\cup\cdots\cup W_k$ and $W_i\neq\emptyset$ for all i.

A set A of edges in a graph $G=(V, E)$ is called a *clique partitioning* of G if there is a partition $\Gamma=\{W_1, \ldots W_k\}$ of V such that $A=E(W_1, \ldots, W_k)$ and such that the subgraph induced by W_i is a clique for $i=1, \ldots, k$. Note that every clique partitioning A induces a unique partition $W_1, \ldots, W_k$ of V such that $A=E(W_1, \ldots, W_k)$. In case G is complete every partition of the node set of G induces a clique partitioning.

A *cycle* $C\subseteq E$ of *length* k is an edge set of the form $\{v_1v_2, v_2v_3, \ldots, v_{k-1}v_k, v_1v_k\}$, where $v_i\neq v_j$ if $i\neq j$. For $k\geq 4$ the set

$$\bar{C} := \{v_iv_{i+2} \mid i=1, \ldots, k-2\}\cup\{v_1v_{k-1}, v_2v_k\}$$

is called the set of *2-chords* of C.

We introduce now some basic concepts from polyhedral theory needed in the sequel—see Schrijver (1986) for a comprehensive treatment of this subject.

Our "universe" will be the vector space $\mathbb{R}^E$, where E is the edge set of a graph.

If $F\subseteq E$ then $\chi^F\in\mathbb{R}^E$ denotes the *incidence vector* of F, that is, $\chi^F_e=1$ if $e\in F$, $\chi^F_e=0$ otherwise. We denote the convex hull of a set $S\subseteq\mathbb{R}^E$ by $\mathrm{conv}(S)$.

A *polytope* P is the convex hull of finitely many points, or equivalently, a bounded set that is the intersection of finitely many halfspaces. An inequality $a^Tx\leq\alpha$ is *valid* with respect to P if $P\subseteq\{x \mid a^Tx\leq\alpha\}$. If $a^Tx\leq\alpha$ is valid with respect to P then $F_a := \{x\in P \mid a^Tx=\alpha\}$ is the face induced by $a^Tx\leq\alpha$. A *facet* of P is a face of P that is contained in no other face of P different from P. Equivalently, a facet F of P is a nonempty face with $\dim(F)=\dim(P)-1$, where $\dim(S)$ denotes the *dimension* of a set S, i.e., the maximum number of affinely independent points in S minus 1. If $P\subseteq\mathbb{R}^E$ has dimension $|E|$ then every facet of P is induced by a valid inequality $a^Tx\leq\alpha$ that is unique up to multiplication by a positive constant. If $a^Tx\leq\alpha$ is valid for P and $F_a := P\cap\{x \mid a^Tx=\alpha\}$ is a facet of P we say that $a^Tx\leq\alpha$ is *facet-defining*.

If $x\in\mathbb{R}^E$ and $F\subseteq E$ then the sum $\sum_{e\in F} x_e$ is abbreviated by $x(F)$. $|S|$ denotes the cardinality of a set S, and, for $\alpha\in\mathbb{R}$, $\lfloor\alpha\rfloor$ denotes the largest integer not larger than α.

2. Clustering, aggregation, clique partitioning

The generic clustering problems we are considering in this paper are most frequently stated in the following (imprecise) form:

> Given n objects and p characteristics and an $n\times p$ data matrix $D=(d_{ik})$, where an entry d_{ik} represents the property of object i with respect to characteristic k, find a "best" partition of the set of objects into nonoverlapping classes of "homogeneous" objects, i.e., find a best clustering of the objects. (2.1)

Some of the terms above need explanation. The data matrix D may contain numerical entries but—in the area of applications we have in mind—it is not the numerical values that are important. We view a column of D as a way to describe which of the objects has which property of the characteristic. For instance, if the objects are "people" and the characteristic is "color of hair" then the entries could be "black", "brown", "red" etc., or if the objects are "voters" and the characteristic is a certain "motion" then one would use the numbers 0, 1, 2 and 3 to indicate whether a voter was in favor, against, abstained, or absent. (So the data are nominal and that is why this area is often called qualitative data analysis.) Another natural way to represent such a problem is, thus, to associate with every characteristic k a binary relation R_k (which represents the "similarity" between the object pairs i, j) by defining

$$(i, j)\in R_k \Leftrightarrow d_{ik}=d_{jk}.$$

Even if exact numerical data are available it is sometimes reasonable to "condense" these data into binary relations. One way, for instance, is to group objects according

to their "size", e.g., states, companies, or mammals are classified as small, medium, or large (or whatever seems appropriate for the application considered).

There are several ways to interpret the terms "best" and "homogeneous" used in (2.1). Having made the step into the language of binary relations it is most natural to say that a "partition of the objects into nonoverlapping classes of "homogeneous" objects" is nothing but an equivalence relation on the set of objects. So the clusters of a clustering are exactly the equivalence classes defined by the equivalence relation. And a natural way to define "best" is to compare each equivalence relation R with each of the given binary relations R_k, i.e., to count the number of disagreements of R with R_k (this is the number $|R \triangle R_k| := |\{(i,j) : (i,j) \in R$ and $(i,j) \notin R_k$, or $(i,j) \notin R$ and $(i,j) \in R_k\}|$), add up the number of disagreements over all given relations, and choose the best equivalence relation with respect to this criterion. Using these interpretations, the clustering problem (2.1) can then be reformulated (precisely) as follows.

> Given p binary relations $R_1, \ldots, R_p$ defined on a set N, find an equivalence relation R^* on N such that $\sum_{k=1}^{p} |R^* \triangle R_k|$ is as small as possible. (2.2)

Problem (2.2) is a classical problem in the area of qualitative data analysis. It is a special type of the so-called *problems of aggregation of binary relations* and has been investigated intensively (cf. Barthélemy and Monjardet (1981), Marcotorchino and Michaud (1980, 1981a, 1981b), Opitz and Schader (1984), and Tushaus (1983)). In the sequel we show how it can be reduced to a 0/1 programming problem.

Every binary relation on a set N can be represented by a 0/1 matrix as follows. For every k, $1 \le k \le p$, and every pair $(i,j) \in N \times N$ we let

$$r_{ij}^{(k)} := \begin{cases} 1 & \text{if } (i,j) \in R_k, \\ 0 & \text{otherwise.} \end{cases}$$

Similarly, we let

$$r_{ij} := \begin{cases} 1 & \text{if } (i,j) \in R, \\ 0 & \text{otherwise.} \end{cases}$$

Then the (nonlinear) objective function $\sum_{k=1}^{p} |R \triangle R_k|$ can be linearized as follows:

$$\begin{aligned} \sum_{k=1}^{p} |R \triangle R_k| &= \sum_{k=1}^{p} \sum_{(i,j) \in N \times N} (r_{ij}^{(k)} - r_{ij})^2 \\ &= \sum_{k} \sum_{(i,j)} r_{ij}^{(k)} + \sum_{k} \sum_{(i,j)} (1 - 2r_{ij}^{(k)}) r_{ij} \\ &= c + \sum_{(i,j)} \left(\sum_{k} (1 - 2r_{ij}^{(k)}) \right) r_{ij} \\ &= c + \sum_{(i,j)} c_{ij} r_{ij}, \end{aligned}$$

That is,

$$\sum_{k=1}^{p} |R \triangle R_k| = \sum_{(i,j) \in N \times N} c_{ij} r_{ij} + c$$

with

$$c_{ij} := \sum_{k=1}^{p} (1 - 2r_{ij}^{(k)}) = p - 2|\{k \in \{1, \ldots, p\} : (i,j) \in R_k\}|$$

and

$$c := \sum_{(i,j) \in N \times N} |\{k \in \{1, \ldots, p\} : (i,j) \in R_k\}|.$$

It is also easy to express the requirement that a binary relation R has to be reflexive, symmetric, and transitive by means of equations and inequalities involving the r_{ij} and hence problem (2.2) can be formulated as

$$\begin{aligned} \text{minimize} \quad & \sum_{(i,j) \in N \times N} c_{ij} r_{ij} + c & & (2.3) \\ \text{subject to} \quad & r_{ii} = 1 \text{ for all } i \in N & & \text{(i.e., } R \text{ is reflexive),} \\ & r_{ij} - r_{ji} = 0 \text{ for all } i,j \in N,\ i \neq j & & \text{(i.e., } R \text{ is symmetric),} \\ & r_{ij} + r_{jk} - r_{ik} \le 1 \text{ for all } i,j,k \in N & & \text{(i.e., } R \text{ is transitive),} \\ & r_{ij} \in \{0,1\} \text{ for all } i,j \in N. & & \end{aligned}$$

Problem (2.3) can be simplified further. Clearly, we can drop the constant c from the objective function and we can also delete the variables r_{ii} from (2.3). Since $r_{ij} = r_{ji}$ we can replace these two variables by one variable x_{ij} (here the order of i and j is irrelevant, so we may assume $i < j$), and by defining new weights $w_{ij} := c_{ij} + c_{ji}$, we get the following 0/1 linear program:

$$\begin{aligned} \text{minimize} \quad & \sum_{1 \le i < j \le n} w_{ij} x_{ij} & & \\ \text{subject to} \quad & x_{ij} + x_{jk} - x_{ik} \le 1 \text{ for all } 1 \le i < j < k \le n, & & \\ & x_{ij} - x_{jk} + x_{ik} \le 1 \text{ for all } 1 \le i < j < k \le n, & & \\ & -x_{ij} + x_{jk} + x_{ik} \le 1 \text{ for all } 1 \le i < j < k \le n, & & (2.4) \\ & x_{ij} \in \{0,1\} \text{ for all } 1 \le i < j \le n, & & \end{aligned}$$

which is clearly equivalent to (2.3).

It is easy to see that the solutions of (2.4) are exactly the incidence vectors of the clique partitionings of the complete graph $K_n = (V_n, E_n)$. Thus the clustering problem (2.1), resp. its related theoretical interpretation (2.2), can be reduced to the following

combinatorial optimization problem (called *clique partitioning problem* (short: CPP)):

> Given a complete graph $K_n = (V_n, E_n)$ with weights $w_e \in \mathbb{Z}$ for all $e \in E_n$, find a clique partitioning $A \subseteq E_n$ such that $w(A)$ is as small as possible. (2.5)

We want to attack the clustering problem (2.2) algorithmically via the clique partitioning problem (2.5). Unfortunately, problems (2.2) and (2.5) are NP-hard, cf. Wakabayashi (1986); so we cannot expect to obtain a polynomial time algorithm. We have chosen to develop a cutting plane algorithm for (2.5). To do this, the polyhedron associated with the clique partitioning problem has to be investigated. This approach will be explained in the following section.

3. The clique partitioning polytope

To formulate CPP in polyhedral, respectively linear programming terms, we associate with it a polyhedron in the following way. Let $\mathbb{R}^{E_n}$ denote the real vector space where every component x_e of a vector $x \in \mathbb{R}^{E_n}$ is indexed by an edge e of the complete graph $K_n = (V_n, E_n)$. To avoid trivialities, we assume throughout the paper that $n \geq 3$. For every edge set $A \subseteq E_n$, $\chi^A \in \mathbb{R}^{E_n}$ denotes its incidence vector. The convex hull of all incidence vectors of clique partitionings of K_n is called the *clique partitioning polytope* (of K_n) and is denoted by $\mathcal{P}_n$, i.e.,

$$\mathcal{P}_n = \text{conv}\{\chi^A \in \mathbb{R}^{E_n} \mid A \text{ is a clique partitioning of } K_n\}.$$

Since the vertices of $\mathcal{P}_n$ are in one-to-one correspondence with the clique partitionings of K_n, it follows immediately that CPP can be formulated as the problem

$$\begin{aligned} &\text{minimize} && w^T x \\ &\text{subject to} && x \in \mathcal{P}_n. \end{aligned} \tag{3.1}$$

(3.1) is a linear program in the sense that a linear objective function is to be minimized over a polytope. To apply LP-techniques, this formulation is of no use unless $\mathcal{P}_n$ can be represented by a system of linear inequalities. Since the clique partitioning problem (2.5) is NP-hard, it follows from general results of complexity theory that it is very unlikely that an explicit complete description of $\mathcal{P}_n$ can ever be obtained. But we were able to determine large classes of valid and facet-defining inequalities for $\mathcal{P}_n$. The following theorem is a summary of some of the results of Grötschel and Wakabayashi (1987).

Theorem 3.2. *Let $K_n = (V_n, E_n)$ be a complete graph with $n \geq 3$ nodes, and let $\mathcal{P}_n \subseteq \mathbb{R}^{E_n}$ be the clique partitioning polytope of K_n.*

(a) *The dimension of $\mathcal{P}_n$ is equal to $|E_n| = n(n-1)/2$.*

(b) *For every edge $e \in E_n$, the trivial inequalities $x_e \geq 0$ and $x_e \leq 1$ are valid for $\mathcal{P}_n$. Every inequality $x_e \geq 0$ defines a facet of $\mathcal{P}_n$ but no inequality $x_e \leq 1$ does.*

(c) *For every three different nodes $i, j, k \in V_n$, each of the three associated triangle inequalities*

$$x_{ij} + x_{jk} - x_{ik} \leq 1, \qquad x_{ij} - x_{jk} + x_{ik} \leq 1, \qquad -x_{ij} + x_{jk} + x_{ik} \leq 1,$$

defines a facet of $\mathcal{P}_n$.

(d) *For every two disjoint nonempty subsets S, T of V_n, the 2-partition inequality induced by S and T (short: $[S, T]$-inequality)*

$$x([S:T]) - x(E_n(S)) - x(E_n(T)) \leq \min\{|S|, |T|\}$$

is valid for $\mathcal{P}_n$. It defines a facet of $\mathcal{P}_n$ if and only if $|S| \neq |T|$.

(e) *For every cycle $C \subseteq E_n$ of length at least 5 and its set $\bar{C}$ of 2-chords, the 2-chorded cycle inequality*

$$x(C) - x(\bar{C}) \leq \left\lfloor \frac{|C|}{2} \right\rfloor$$

is valid for $\mathcal{P}_n$. It defines a facet of $\mathcal{P}_n$ if and only if $|C|$ is odd.

(f) *For every even cycle $C \subseteq E_n$ of length at least 8, for every node $z \in V_n$ not in the node set $V_n(C)$ of C, and for every bipartition V, $\bar{V}$ of $V_n(C)$, the 2-chorded even wheel inequality*

$$x(C \cup R) - x(\bar{C} \cup \bar{R}) \leq \frac{|C|}{2}$$

defines a facet of $\mathcal{P}_n$, where $\bar{C}$ is the set of 2-chords of C and $R := \{zv \mid v \in V\}$, $\bar{R} := \{zv \mid v \in \bar{V}\}$. □

The proofs of these results are quite involved. These are not all facets of $\mathcal{P}_n$ known—see Grötschel and Wakabayashi (1987) for further details. Let us set

$$\mathcal{T}_n := \{x \in \mathbb{R}^{E_n} \mid x \geq 0, x \text{ satisfies all triangle inequalities 3.2 (c)}\}.$$

Then, by (3.2), all inequalities defining $\mathcal{T}_n$ induce facets of $\mathcal{P}_n$ and, by (2.4),

$$\mathcal{P}_n = \text{conv}\{x \in \mathcal{T}_n \mid x \text{ integral}\}.$$

So the linear program

$$\begin{aligned} &\text{minimize} && w^T x \\ &\text{subject to} && x \in \mathcal{T}_n, \end{aligned} \tag{3.3}$$

is an LP-relaxation of the clique partitioning problem. Our computational experiments—see Section 5—show that (3.3) is indeed quite a reasonable LP-relaxation.

4. The cutting plane algorithm

The use of polyhedral results in LP-based cutting plane procedures has become a standard (and very successful) technique in the recent years—see, for instance,

Barahona, Grötschel, Jünger and Reinelt (1988), Crowder and Padberg (1980), Grötschel and Holland (1985), Grötschel, Jünger and Reinelt (1984), Padberg and Rinaldi (1987), Reinelt (1985). Nevertheless, to make the basic idea work, requires some nontrivial, rather problem dependent effort.

We will first describe the fundamentals of our algorithm. Afterwards we go into more detail and explain some implementational aspects and some of the heuristics we have added to the basic algorithm to make it work in practice.

The initial step of the polyhedral approach to combinatorial optimization is to get a very good linear programming relaxation of the problem considered. We believe—and computational experience shows—that the system of inequalities given by the classes of facets described in Theorem 3.2 is such an LP relaxation. It is, however, far from clear how linear programs over these systems of inequalities can be solved efficiently.

It follows from the ellipsoid method, see Grötschel, Lovász and Schrijver (1981), that a linear program (with possibly exponentially many inequalities) can be solved in polynomial time if the separation problem for the constraint system can be solved in polynomial time. The separation problem here is the task to check, for a given vector y, whether y satisfies all constraints, and if not, to find a constraint that is violated by y. (Such a constraint is called a *cutting plane* or *cut*, for obvious reasons.)

For the classes of facet-defining inequalities given in Theorem 3.2, it is clear that the trivial constraints (b) and the triangle inequalities (c) can be checked in polynomial time. However, we do not know whether the separation problem for any of the other systems of inequalities can be solved in polynomial time. Thus to take care (at least partly) of these inequalities we have to resort to separation heuristics.

Due to these facts we decided to proceed as follows. We first solve (by trivial inspection) the linear program

$$\text{minimize} \quad w^T x$$
$$\text{subject to} \quad 0 \le x_e \le 1 \text{ for all } e \in E_n. \tag{4.1}$$

Suppose x^* is an optimum solution of the present LP. We now check whether x^* violates any of the triangle inequalities. If so, we add some of these inequalities (the selection process will be described later) to the current LP, and we repeat.

If x^* satisfies all triangle inequalities and is integral we know that x^* is the incidence vector of a clique partitioning and stop with an optimum solution of (4.1).

If x^* satisfies all triangle inequalities and is fractional there is no obvious way to continue, since no further polynomial time separation algorithm is available. We have thus invented several heuristics that check whether a given point x^* violates inequalities of the other classes described in Theorem 3.2. Our final choice of which separation heuristic to use was then based on extensive computational testing. This is not too satisfying from a theoretical point of view, but we do not know a way to get around this kind of numerical experiments.

The result of these experiments was that we gave up considering the 2-chorded cycle inequalities and the 2-chorded odd wheel inequalities (and the further facets we know) completely. We only added three heuristics that search for violated 2-partition inequalities 3.2 (d).

So our algorithm now calls these three heuristics to check whether some 2-partition inequalities violated by x^* can be found. If this is so, these inequalities are added to the current LP and we repeat. Otherwise, we resort to branch and bound.

It was more than surprising for us that, in all problems we ran, we never had to call the branch and bound algorithm. The cutting plane phase always ended with an integral optimum solution. So the chosen LP-relaxation of (3.1) consisting of all inequalities 3.2 (b) and (c) and some of the inequalities 3.2 (d) turned out to be (empirically) very effective.

Let us now describe a few more details to show our strategic and tactical choices.

LP-solver: Of couse, we did not use the ellipsoid method. We solved our linear programs with IBM's LP-package MPSX/370. This is quite a fast LP-solver, but it is not so easy to use it in a cutting plane environment. Anyway, MPSX did its job. The very first LP was solved by inspection. In all subsequent problems we first called the routine DUAL to find a feasible solution (based on the optimum basis of the previous LP), and then we called PRIMAL to obtain an optimum solution.

Variable elimination: The first issue to think of is whether one tries to eliminate variables in order to run a sparse subproblem. In this case one has to write a routine that brings in the left out variables, prices them out, and restarts the whole program on a larger set of variables in case some reduced costs have the wrong sign. The largest real world problem we got is a clustering problem for 158 objects. So this gives a clique partitioning problem with 12 403 variables. This size is around the break even point where pricing out as described above starts to pay (an observation made in other cutting plane experiments). Thus we decided to drop that option and run on the full variable set.

Checking triangle inequalities: There are $3\binom{n}{3}$ triangle inequalities 3.2 (c). Based on the problem dimensions we were going for, we decided that complete enumeration of all the inequalities would be feasible with respect to running time. It turned out, however, that, by checking all triangle inequalities, sometimes several thousand violated inequalities were found. To keep the LP's small we decided to add not all of them to the current LP. After some experiments (more on that in Section 5) we adopted the following strategy. First, we introduced a parameter, called MAXCUT, to limit the number of cutting planes added in one iteration. In addition, the enumeration process was organized in such a way that, as soon as about $5\times$ MAXCUT violated triangle inequalities are found, the enumeration terminates. Moreover, for each violated triangle inequality $a^T x^* \le 1$ we compute its "degree of violation", i.e., the number $\delta_a := a^T x^* - 1$, and sort these numbers into 10 buckets

of equal size. After termination of this procedure we start retrieving violated triangle inequalities from the buckets (starting with the bucket of highest degree of violation) until MAXCUT inequalities have been chosen or all buckets are empty. This way we generate a "reasonable" number of cutting planes all of which are "highly" violated. Limited computational experiments showed that

$$\text{MAXCUT} \in \{400, 500\}$$

is a good choice for our range of problem sizes.

Note that by considering triangle inequalities first and handling them in the way described above yields (empirically) the following benefits:

- The LP's are kept small and sparse (with considerable numerical and running time advantages).
- The heuristics for other classes of inequalities can use the fact that all triangle inequalities are satisfied.

Separation heuristics for 2-partition inequalities: Here we assume that all triangle inequalities are satisfied. As before, we will not generate more than MAXCUT cutting planes in one phase.

We first run a heuristic that searches for $[S, T]$-inequalities with $|S| = 1$. For every node $v \in V_n$ we do the following. We set $W := \{w \in V_n \setminus \{v\} \mid 0 < x^*_{vw} < 1\}$ (the nodes with $x^*_{vw} \in \{0, 1\}$ will not help in this process) and choose some ordering of the nodes of W. We pick the first node $w \in W$ (in this ordering), set $T := \{w\}$, and for every node $i \in W \setminus \{w\}$ we set

$$T := T \cup \{i\} \quad \text{if } x^*_{ij} = 0 \text{ for all } j \in T. \tag{4.2}$$

We check whether the set T constructed this way satisfies $x^*([\{v\} : T]) > 1$. If so the inequality $x([\{v\} : T]) \leq 1$ is added to the current LP. We repeat this procedure with converse ordering of W.

We call the next heuristic only if the previous one failed to produce any violated inequality. The second heuristic is the same as the first, we only replace (4.2) by

$$T := T \cup \{i\} \quad \text{if } x^*_{iw} - \sum_{j \in T} x^*_{ij} > 0. \tag{4.3}$$

Both these heuristics have an $O(n^3)$ worst case running time but are much faster in practice.

Our third routine searches for violated $[S, T]$-inequalities with $|S|, |T| \geq 2$. It is a complex enumerative procedure with $O(n^3)$ running time, and we do not want to describe it here. In fact, due to our order of calling the heuristics, it turned out that this last heuristic was never ever called in our applications.

Row elimination: After having determined some new cutting planes and before restarting the LP-solver we eliminate all old constraints that are nonbinding at the current optimum solution, i.e., we delete all those rows $a^T x \leq \alpha$ with $a^T x^* < \alpha$. It may, thus, happen that some cuts are generated several times; but in general, row elimination tends to produce smaller linear programs and an overall running time decrease.

Branch and Bound: If the cutting plane phase does not produce an integral solution a branch and bound procedure is called to find an optimum solution. As the implementation of branch and bound algorithms is well known we shall not elaborate on this. We only want to mention that when a variable is temporarily (resp. permanently) fixed to 1 or 0 then we can use the triangle inequalities to fix temporarily (resp. permanently) other variables. That is, if x_a and x_b have been fixed to 1 and $\{a, b, c\}$ is a triangle in K_n, then we can fix x_c to 1. This amounts to finding the *transitive closure* of the graph defined by the edges e with $x_e = 1$. As stated before, the branch and bound phase was never activated in the final version of our algorithm.

5. Applications and computational results

In this section we report on the computational experiences with our cutting plane algorithm for the clique partitioning problem. This algorithm was primarily developed for the clustering problem (2.1) resp. (2.2) which is—as we showed in Section 2—reducible to CPP. All real world instances of CPP we consider here arise this way.

As mentioned in Section 2, the p binary relations R_k, $1 \leq k \leq p$, on the object set $N := \{1, 2, \ldots, n\}$ are defined by

$$(i, j) \in R_k \Leftrightarrow d_{ik} = d_{jk}.$$

Thus the corresponding instance of CPP consists of a complete graph $K_n = (V_n, E_n)$ with weights w_{ij} assigned to its edges, where $w_{ij} := p - 2|\{k \in \{1, \ldots, p\} : (i, j) \in R_k\}|$. In two of the applications to be mentioned in the next section some binary relations (that are based on quantitative data) will be defined in a slightly different way.

In all applications given here the input to our program is the data matrix D whose columns define the given binary relations and an additional parameter which determines how the weights w_{ij} are to be calculated. The output is a list of the objects in each of the equivalence classes (i.e., a list of the clusters) determined by the optimum clique partitioning.

We ran our program on a Siemens 7.865 of the Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt (DFVLR) in Oberpfaffenhofen under the operating system VM/370-CMS. The CPU time reported here is the figure printed out by the operating system as the execution time for the entire run, including all input and output operations. The fractions of seconds were rounded up.

We have considered many real as well as random data but—as mentioned before—in none of the cases we needed to go into the branch and bound phase. More surprisingly, in most cases triangle inequalities were sufficient to produce an

integral optimum solution. Only a few times 2-partition inequalities (with $|S| = 1$) had to be added.

5.1. The problem instances and the optimization process

In the sequel we consider various examples of clique partitioning problems to illustrate the performance of our algorithm on real world problem instances. Most of the data sets correspond to applications from the literature concerning the problem of classifying objects based on their qualitative and/or quantitative description. Some new data sets that have not yet appeared in the literature are also given. The complete data sets of all problems considered, and the corresponding solutions are given in the Appendix.

For each problem instance we have summarized the relevant execution data in a table. All Tables 5.1–5.13 are printed in the same scheme. The symbol "#" means "number of". The optimization process is shown step by step and the total CPU time for the entire run is given. The column "Iter." indicates the iteration number, where each iteration is one complete cutting plane phase (cutting plane generation, row elimination, solution of the new LP). Thus, the last number in this column is the number of LP's that had to be solved. The next three columns show how many cuts are generated, added, and eliminated per iteration. In the column "LP size" we give the number of constraints of the current LP – the one obtained after the addition and elimination of the cuts indicated in the previous columns. The last column shows the objective function value for the optimum solution of the current LP. The last value in this column is the objective function value for the optimum integer solution found. As it will be clear from the tables, we have considered MAXCUT = 400 or MAXCUT = 500.

To distinguish between the types of cuts generated, we use the symbol "A" to denote triangle inequalities and "*" to denote $|S, \bar{S}|$-inequalities with $|S| = 1$.

Note that the first row (Iter. = 0) corresponds to the trivial LP-14-1 whose optimum solution is obvious. The cuts generated in iteration k, $k \geq 1$, are based on the optimum solution obtained in iteration $k - 1$.

5.1.1. Classification of cetacea (whales, porpoises, and dolphins)

The problem of classifying animals and plants is of great interest in biology. The first example we want to consider here was proposed by Vescia (1985) and concerns the classification of 36 different types of cetacea. The given instance consists of 35 different genera and one species (the Balaenoptera musculus) which are described with respect to 15 characteristics (10 morphological, 3 osteological and 2 behavioral parameters).

Many different mathematical models, including the one we used, have been proposed for this particular problem instance and slightly different classifications were found. As a matter of fact, the cetacea are still poorly known and different zoologists propose different classifications.

The computational results we obtained with the application of our cutting plane algorithm to this data set ($n = 36$, $p = 15$, # LP variables = 630) are summarized in Table 5.1.

Table 5.1

Iter.	# Cuts Generated	Added	Eliminated	LP size	Obj. value
0					998
1	548A	548		274	967
2	12A	12		286	967

CPU time: 0.15 (min:sec)

The information given in Table 5.1 are to be interpreted as follows: At first the trivial LP was solved and an optimum solution, say x^0, was obtained. The objective function value at x^0 was found to be 998. Then, 274 triangle inequalities which were violated by x^0 were generated, and these were all added to the trivial LP. The new LP with 274 constraints was solved and an optimum solution x^1 with $w^t x^1 = 967$ was obtained. Twelve triangle inequalities violated by x^1 were generated and these were added to the previous LP yielding a new LP with 286 constraints. The optimum solution of this last LP, with objective function value 967, was integral and an optimum solution for CPP.

This problem instance has been solved to optimality by Vescia (1985), using the approach of solving the dual version of the LP defined by the triangle inequalities and the nonnegativity constraints. The author, however, does not report any computational details.

Remark. In this particular problem some information is missing (i.e. entries "*" in the Table A1 given in the Appendix A1) and for this reason the weights w_{ij}, $i < j$, were calculated as follows:

$$w_{ij} := s_{ij} - s_{ij}$$

where

$$s_{ij} := |\{k:\ d_{ik} = d_{jk},\ d_{ik} \neq *,\ 1 \leq k \leq p\}|$$

and

$$s_{ij} := |\{k:\ d_{ik} \neq d_{jk},\ d_{ik} \neq *,\ d_{jk} \neq *,\ 1 \leq k \leq p\}|.$$

It is assumed here that when an entry, say d_{ik}, in the data matrix is "*" then the object i is not comparable to any of the other objects with respect to the characteristic k. Thus s_{ij} (resp. s_{ij}) corresponds to the number of characteristics with respect to

which the objects i and j have (resp. do not have) the same description, provided they are comparable. Note that in this case $s_{ij} + s_{ij} = p$, but if in particular $s_{ij} + s_{ij} = p$, then $w_{ij} = \bar{s}_{ij} - s_{ij} = p - 2s_{ij}$. That is, when no information is missing we have our usual definition.

5.1.2. Classification of wild cats

This problem instance is similar to the previous one. The data set consists of the description of 30 wild cats with respect to 14 morphological and behavioral characteristics (cf. Appendix A2). This example is from Marcotorchino (1981), who uses the same mathematical model but solves it to optimality with another method. Again only two iterations were needed to find an optimal solution and only 555 cutting planes sufficed to prove optimality. (In this instance $n = 30$, $p = 14$ and # LP variables = 435).

Table 5.2

	# Cuts				
Iter.	Generated	Added	Eliminated	LP size	Obj. value
0	...				1400
1	561 Δ	400		400	1313
2	155 Δ	155		555	1304

CPU time: 0:23 (min:sec)

5.1.3. Classification of workers

In this example from Opitz and Schader (1984) the data matrix (given in Appendix A3) describes the opinions of 34 workers with respect to 13 parameters concerning their working environment, colleagues, headman, career perspectives, etc.

We have been informed (private communication) that Opitz and Schader have solved this problem to optimality using a Lr and r bound method proposed by Tüshaus (1983). (In this example $n = 34$, $p = 13$ and # LP variables = 561).

Table 5.3 shows that—contrary to the previous examples—violated inequalities other than the triangle inequalities had to be added to find an optimum solution. The solution found at iteration 7 satisfies all triangle inequalities but is fractional. Two violated simple 2-partition inequalities were generated at iteration 8 and these were added to the previous LP. The new LP, with 1138 constraints, yielded an optimum solution for CPP. In fact, among the real world problems, this is the only example where inequalities different from triangle inequalities were needed.

5.1.4. Classification of cars

In this problem instance 33 cars are described with respect to the frequency repair that was needed for the brake system, fuel system, etc. A total of 11 parameters are

Table 5.3

	# Cuts				
Iter.	Generated	Added	Eliminated	LP size	Obj. value
0					1233.0
1	1849 Δ	400		400	1168.50
2	1481 Δ	400		800	1074.50
3	1268 Δ	400	173	1027	1012.83
4	1751 Δ	400	344	1083	970.00
5	89 Δ	89	39	1103	965.50
6	41 Δ	41	10	1097	964.50
7	66 Δ	66	26	1137	964.50
8	2 *	2	1	1138	964.00

CPU time: 0:18 (min:sec)

considered (cf. Appendix A4) and the entries are 0 and 1 (where "1" stands for greater than average frequency of repair). This data set is given in Hartigan (1975) as a trial data set for an algorithm to construct a *tree*—a special clustering structure. The results are therefore, not comparable. (We have $n = 33$, $p = 13$ and # LP variables = 528).

Table 5.4

	# Cuts				
Iter.	Generated	Added	Eliminated	LP size	Obj. value
0					1748.00
1	1807 Δ	400		400	1648.00
2	1264 Δ	400		800	1587.50
3	751 Δ	400	55	1145	1502.00
4	40 Δ	40	17	1158	1501.00

CPU time: 5:45 (min:sec)

5.1.5. Classification of micro computers

The data set considered here is from Chuh (1985) and concerns the description of 40 micro computers with respect to 14 characteristics (see Appendix A5).

To deal with some quantitative values (describing price, size of random access memory, etc.) in this particular case we defined the binary relations R_k, $1 \le k \le 14$, as follows:

$$(i, j) \in R_k \Leftrightarrow \frac{|d_{ik} - d_{jk}|}{\max\{d_{ik}, d_{jk}, 1\}} \le 0.3.$$

The model used by Chuh is different from the one we use, and leads to a different result.

Table 5.5

	# Cuts				
Iter.	Generated	Added	Eliminated	LP size	Obj. value
0	—	—			1570
1	1575 Δ	500		500	1158
2	1434 Δ	500		1000	1065
3	1185 Δ	500	279	1221	996
4	1297 Δ	500	305	1416	966
5	42 Δ	42	116	1342	966

CPU time: 4:17 (min:sec)

5.1.6. *Votes of 54 member states of the United Nations Organization*

The example considered here, given by Marcotorchino (1981), describes the votes of 54 Nations on 3 motions presented in a General Assembly of the UNO in 1968 (see Appendix A6). This problem has been solved to optimality by Marcotorchino, using the approach of solving the dual version of the LP relaxation defined by the triangle and trivial inequalities.

Table 5.6

	# Cuts				
Iter.	Generated	Added	Eliminated	LP size	Obj. value
0	—	—			918 180
1	2004 Δ	400		400	866 000
2	2001 Δ	400		800	840 180
3	2003 Δ	400	281	919	811 000
4	1716 Δ	400	70	1249	801 000
5	19 Δ	19		1268	798 180

CPU time: 4:30 (min:sec)

5.1.7. *Votes of all member states of the United Nations Organization*

These are new data sets, collected from UNO (1985), concerning the votes of all member states of the United Nations Organization on *Resolutions adopted by the 39th General Assembly* held at the end 1984 (cf. Appendix A7).

For each of the data sets to be specified in the sequel, we have run our program for the complete data set ($n = 158$) and for a smaller data set ($n = 158$) obtained from the complete one by deleting some States which were "absent".

Case 1: Resolutions 39/119-120-121—Votes on 3 Resolutions concerning the situation of human rights and fundamental freedom in *El Salvador, Guatemala, and Chile* (cf. Appendix A7.1).

Table 5.7

	# Cuts				
Iter.	Generated	Added	Eliminated	LP size	Obj. value
0	—	—		—	12 322
1	2001 Δ	400		400	12 298
2	2001 Δ	400		800	12 280
3	2001 Δ	400	30	1170	12 234
4	1681 Δ	400		1570	12 210
5	1402 Δ	400	37	1933	12 207
6	1401 Δ	400		2333	12 197

CPU time: 14:12 (min:sec)

Case 1a: $n = 158$ and $p = 3$ (# LP variables = 12 403). See Table 5.7.

Case 1b: $n = 139$ and $p = 3$ (# LP variables = 9591). This data set corrsponds to the previous one without the States which were absent in at least one of the three sessions consitdered. See Table 5.8.

Table 5.8

	# Cuts				
Iter.	Generated	Added	Eliminated	LP size	Obj. value
0	—	—		—	11 706
1	2008 Δ	400		400	11 664
2	2001 Δ	400		800	11 650
3	2005 Δ	400	40	1160	11 639
4	846 Δ	400		1560	11 613

CPU time: 8:38 (min:sec)

Table 5.9

	# Cuts				
Iter.	Generated	Added	Eliminated	LP size	Obj. value
0	—	—		—	73 178
1	2502 Δ	500		500	73 139
2	2502 Δ	500		1000	72 886
3	2501 Δ	500	166	1334	72 874
4	236 Δ	236		1570	72 821
5	87 Δ	87	33	1624	72 820

CPU time: 9:47 (min:sec)

Case 2: Resolution 39/148—Votes on 15 matters related with nuclear weapons (cf. Appendix A7.2).

Case 2a: $n = 158$ and $p = 15$ (# LP variables = 12 403). See Table 5.9.

Case2b: $n = 145$ and $p = 15$ (# LP variables = 11 165). The data set corresponds to the previous one without the States which were absent at least 8 times. See Table 5.10.

Case 3: Resolution 39/99—Votes on 9 matters related to the *United Nations Relief and Works for Palestine Refugees in the Near East* (cf. Appendix A7.1).

Case 3a: $n = 158$ and $p = 9$ (# LP variables = 12 403). See Table 5.11.

Table 5.10

	# Cuts				
Iter.	Generated	Added	Eliminated	LP size	Obj. value
0	...				[illegible]
1	2506 Δ	500		500	[illegible]
2	2513 Δ	500		1000	[illegible]
3	1426 Δ	500		1500	[illegible]
4	269 Δ	269		1969	[illegible]
5	87 Δ	87	98	1858	[illegible]

CPU time: 8:04 (min:sec)

Table 5.11

	# Cuts				
Iter.	Generated	Added	Eliminated	LP size	Obj. value
0	...				[illegible]
1	2502 Δ	500		500	[illegible]
2	2501 Δ	500		1000	[illegible]
3	840 Δ	500		1500	[illegible]

CPU time: 5:27 (min:sec)

Table 5.12

	# Cuts				
Iter.	Generated	Added	Eliminated	LP size	Obj. value
0	...				[illegible]
1	2502 Δ	500		500	[illegible]
2	2501 Δ	500		1000	[illegible]
3	840 Δ	500		1500	[illegible]

CPU time: 4:43 (min:sec)

Case 3b: $n = 147$ and $p = 9$ (# LP variables = 7 731). This data set corresponds to the previous one without the States which were absent at least 5 times. See Table 5.12.

5.1.8. Classification of companies

The problem instance considered here is from Späth (1977) and concerns the description of 137 companies in W. Germany with respect to their need of different groups of employees (cf. Appendix A8). A total of 25 groups of employees are considered (ex. secretaries, programers, engineers, etc.), and for each company, it is indicated whether the corresponding group of employees is needed ("1") or not ("0"). The clustering technique used by Späth fixes the number of clusters a priori and optimizes a different objective function. The computational results are therefore not comparable.

Table 5.13

	# Cuts				
Iter.	Generated	Added	Eliminated	LP size	Obj. value
0					82 625
1	2501 Δ	500		500	82 414
2	2502 Δ	500		1000	82 188
3	2565 Δ	500	182	1318	82 163
4	2550 Δ	500	131	1687	82 051
5	2502 Δ	500	31	2156	81 897
6	2501 Δ	500	51	2605	81 884
7	2549 Δ	500		3105	81 879
8	2558 Δ	500	86	3619	81 811
9	1885 Δ	900		4019	81 802

CPU time: 19:47 (min:sec)

Although the optimum solution we obtained turned out to be not interesting (only two clusters were produced), the computational results reveal some interesting aspects (cf. Table 5.13). Note that the LP at iteration 8 has 3619 constraints, all of which are binding for the obtained solution x (this can be derived from the fact that at iteration 9 no elimination was performed). Although there were 1885 violated inequalities violated by x the addition of only 900 of them was sufficient to produce an optimum solution. It should also be noted that in this case the final LP has more than 4000 constraints. This problem is the largest "real world problem we encountered so far.

5.2. Random data

In the applications we considered previously, only in one case ($n = 34$) we needed to add cuts other than the triangle inequalities. For randomly generated data sets,

which we created to test various features of our code the picture was, however, different.

We generated 20 matrices with entries 0, 1, 2, having dimension $n \times p$ where $12 \leq n \leq 30$, $10 \leq p \leq 13$, and in 9 cases we needed to call the cut generation routine for $[S, T]$-partition inequalities with $|S| = 1$. In 7 cases the first strategy used in this routine was sufficient to find violated $[S, T]$-partition inequalities, and in 2 of them we had to use the second strategy (cf. Section 4). In none of the cases we needed to call the third cut generation heuristic for $[S, T]$-inequalities and in all cases (except in one) the running times were less than 2 minutes.

In Table 5.14 we show the computational result obtained for $n = 22$ and $p = 13$. Observe that when all triangle inequalities are satisfied the objective function value (−138.50) is very close to the optimum objective value (−137). This fact could be observed in all the 7 cases (only in one case the relative difference was 5%, in 6

Table 5.14
$n = 22$

	# Cuts Generated			
Iter.	Δ	*(1)	*(2)	Obj. value
0	—	—	—	−237.00
1	432	—	—	−138.50
2	—	2	—	−138.00
3	62	—	—	−138.00
4	—	1	—	−137.75
5	—	1	—	−137.33
6	3	—	—	−137.17
7	—	1	—	−137.11
8	—	3	—	−137.00
9	9	—	—	−137.00
10	—	—	2	−137.00
11	13	—	—	−137.00

CPU time: 1:21 (min:sec)

Table 5.15
$n = 30$

Data	Entries	Cuts	MAXCUT	CPU time
Random-1	{0, 1, 2}	Δ *	400	0:23
Random-2	{0, 1, 2}	Δ *	400	0:15
Random-3	{0, 1, 2}	Δ	400	3:46
Random-4	{0, 1, 2, 3}	Δ *	400	0:11
Random-5	{0, 1}	Δ	400	4:20
Random-6	{0, 1}	Δ	400	2:40

cases it was less than 2%). The symbols "*(1)" and "*(2)" stand for $[S, T]$-inequalities found by the first and second strategy, respectively.

The computational results obtained for 6 data matrices of dimension $30 \times p$, where $10 \leq p \leq 13$, are shown in Table 5.15.

5.3. Some remarks

We have tested to which extent the choice of the value for MAXCUT influences the overall running time and we have noted that in most of the cases the better performances were achieved for MAXCUT $\in \{400, 500\}$, for n up to 60. Especially when there are many violations the running times may depend largely on the choice of MAXCUT. In these cases, if MAXCUT is less than 300 the improvement on the objective value is very slow. On the other hand, if MAXCUT is too large (say more than 800) the LP's become large very fast, even with eliminations, and they require a lot of time to be optimized. In Table 5.16 we indicate the results obtained for $n = 34$ (1849 violations found in the first iteration) and $n = 33$ (1507 violations).

Table 5.16

$n = 34$ (workers)		$n = 33$ (cars)	
MAXCUT	CPU time	MAXCUT	CPU time
250	4:40	300	2:45
400	3:18	400	2:35
500	3:30	500	3:25
600	4:13	800	3:36

By comparing the running times for problem instances with about the same size we observed considerable variances. A closer look at the "structures" of the objective function showed that "easy" problems often have the property that more than 80% of the objective function coefficients are positive (or negative), while the problems with a more even distribution of positive and negative objective function coefficients needed more time for their solution. More exactly, what matters is the distribution of the w_{ij} in the interval $[-p, p]$. But we do not have sufficient material to derive significant statistical conclusions from our impression.

In Table 5.17 we summarize the computational results obtained for all real data and some random data we have mentioned previously. It is an interesting fact that for the real data sets, except for the case $n = 34$, the triangle inequalities were sufficient to obtain an optimum solution. In all cases in which we needed to add cuts other than the triangle inequalities, we obtained that the LP-relaxation defined by the triangle inequalities produced a very good lower bound for the optimum objective value is very slow. On the other hand, if MAXCUT is too large (say more

Table 5.17

n	Data	type of cuts	# iter	max LP size	max # cuts	LP-time (min sec)
12	Random	Δ +	4	15	29	0 11
15	Random	Δ	7	122	337	0 46
20	Random	Δ +	6	361	879	0 51
25	Random	Δ	7	660	741	1 45
30	Random-1	Δ +	5	145	94	0 29
30	Random-2	Δ +	4	96	57	0 17
30	Random-3	Δ	4	955	1016	3 46
30	Random-4	Δ +	3	31	31	0 11
30	Random-5	Δ	4	945	1406	4 20
30	Random-6	Δ	4	1506	1578	5 47
30	Wild Cats	Δ	5	555	561	0 53
33	Cars	Δ	4	1158	1507	5 35
34	Workers	Δ +	8	1148	1849	5 18
36	Cetacea	Δ	5	280	274	0 45
40	Micro	Δ	7	1145	1575	8 17
54	UNO	Δ	5	1398	2004	8 30
60	Random	Δ +	16	915	745	6 05
158	UNO-1a	Δ	6	2311	2003	14 15
139	UNO-1b	Δ	4	1360	2005	6 46
158	UNO-2a	Δ	5	1654	2502	9 47
145	UNO-2b	Δ	5	1818	2513	6 04
158	UNO-3a	Δ	4	1500	2503	5 27
147	UNO-3b	Δ	4	1500	2503	4 33
137	Companies	Δ	9	4019	2556	19 47

branch and bound phase, if needed. Surprisingly, in none of the problem instances we needed to enter the branch and bound phase. In fact, we never called our third cut generation routine for general 2-partition inequalities.

The present form of our algorithm should not be seen as a definitive one. It would be interesting to test some other strategies to see whether the running time can be speeded up. Thus for example in the case of the violated triangle inequalities, instead of using our MAXCUT most violated inequalities strategy, one could try to add only variable disjoint triangle inequalities.

Another possibility would be to consider again some other classes of facet defining inequalities, develop better separation heuristics and call them in different orders. These features could be added to the present code without requiring any substantial modification.

5.4. Conclusions

Other codes for the solution of the clustering problem considered here exist – see Marcotorchino and Michaud (1980, 1981a, 1981b), Schader and Tushaus (1985), or Tüshaus (1983). Unfortunately, we were not able to get hold of any of these codes. So we were not able to execute these algorithms in the same environment in order to compare running times, etc. It is also very hard to draw conclusions from the published results of other authors since the performances of the codes are often not documented too well and, even if so, it is unclear how to filter out the performance characteristics of the computers and operating systems used. Anyway, we believe that our approach is a valid alternative to the existing methods. This opinion, in particular, is confirmed by the fact that the running times of our code are quite modest and that it can handle large problem sizes consistently well. Moreover, as far as we know no other code has ever solved clustering problems of the size we report about here.

Appendix

We list in the sequel either the complete data sets or provide references to papers where the data of the problem instances mentioned in Chapter 5 can be found. The optimum solution we found is listed under the heading "Solution Classes" (SC), where we indicate the class to which each object belongs. Thus, in the column "SC", objects with the same number are to be interpreted as belonging to the same class.

Appendix A1. Classification of cetacea

Reference: Vescia (1985). The parameters and entries in Table A1 are specified as follows:

(i) Morphological parameters

1 *Neck*
0: does not exist, 1: exists.

2 *Form of the head*
0: cylindrical, 1: conical, 2: with a curved forehead, 3: globular, 4: flat, 5: convex.

3 *Size of the head*
0: very big, 1: medium size.

4 *Beak*
0: missing, 1: large, 2: narrow and short, 3: narrow and long.

5 *Dorsal fin*
0: missing, 1: triangular, 2: falciform, 3: backward and falciform.

6 *Flippers*
0: small, 1: large and short, 2: medium size, 3: long and narrow.

7 *Set of teeth*
0: on the lower jaw, 1: on the lower and upper jaw, 2: without teeth but long baleens, 3: without teeth but thick baleens, 4: without teeth but large baleens.

9 *Blow hole*
0: on the left side, 1: on the right side, 2: on the middle line, 3: on the middle line with two holes.

10 *Color*
0: central parts are clearer than dorsal parts, 1: blackish, 2: no pigmentation, 3: spotted.

14 *Longitudinal furrows on the throat*
0: do not exist, 1: a small number exists, 2: a big number exists.

(ii) Osteological parameters

11 *Cervical vertebrae*
0: free, 1: partly or completely welded.

12 *Lachrymal and jugal bones*
0: form one piece, 1: are independent.

15: *Head bones.*
0: symmetrical, 1: slightly unsymmetrical, 2: unsymmetrical, 3: very unsymmetrical

(iii) Behavioral parameters.
8: *Feeding.*
0: feed on squish, 1: feed on fish, 2: feed on seal, 3: feed on plankton
13: *Habitat.*
0: rivers, 1: temperate or warm seas, 2: cold seas, 3: coasts, 4: variable

Table A1
Data matrix and solution classes (SC)

	Zoological name	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	SC
1	Balaena	0	5	0	0	0	1	2	1	1	1	1	*	*	0	0	1
2	Balaenoptera	0	4	0	0	1	0	1	1	1	0	0	*	3	2	0	2
3	Balaenoptera Mus.	0	4	0	0	1	1	1	1	1	1	0	*	3	*	0	2
4	Berardius	1	2	1	2	2	0	0	0	2	0	1	1	2	1	2	0
5	Cephalorhynchus	0	1	1	1	2	2	1	0	1	1	1	1	2	0	1	1
6	Delphinapterus	1	3	1	0	0	1	1	1	1	2	0	1	2	0	2	2
7	Delphinus	0	2	1	2	2	2	1	1	1	0	1	1	1	0	1	1
8	Eschrichtius	0	1	0	0	0	1	1	1	1	1	0	*	1	1	0	2
9	Eubalaena	0	5	0	0	0	1	2	1	0	1	1	*	*	0	0	1
10	Globicephala	0	3	1	0	2	1	1	0	1	1	1	1	1	0	1	1
11	Grampus	0	3	1	0	2	1	0	0	1	0	1	1	1	0	1	1
12	Hyperoodon	1	2	1	2	2	0	0	0	2	0	1	1	1	1	2	1
13	Inia	1	2	1	1	1	1	1	1	2	0	0	0	0	0	*	0
14	Kogia	0	0	0	0	1	0	0	0	0	0	1	*	1	0	1	2
15	Lagenorhynchus	0	2	1	2	2	2	1	1	1	0	1	1	2	0	1	1
16	Lipotes	1	2	1	1	1	1	1	1	0	0	0	0	0	0	*	0
17	Lissodelphis	0	1	1	1	0	2	1	1	1	0	1	1	1	0	1	1
18	Megaptera	0	4	0	0	1	1	1	1	1	1	0	*	1	2	0	2
19	Mesoplodon	1	1	1	2	2	0	0	0	2	0	1	1	1	1	2	1
20	Monodon	1	3	1	0	0	1	0	0	1	1	0	1	2	0	2	2
21	Neophocaena	0	1	0	0	1	0	2	1	1	0	1	*	2	0	0	1
22	Neophocaena	0	1	1	0	0	0	1	1	1	1	1	1	1	0	1	1
23	Orcaella	1	3	1	0	2	0	1	1	1	1	1	1	1	0	1	1
24	Orcinus	0	3	1	0	2	1	1	2	1	1	1	1	2	0	1	1
25	Phocaena	0	1	1	0	1	0	1	1	1	0	1	1	1	0	1	1
26	Physeter	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	2
27	Platanista	1	2	1	1	1	1	1	1	2	1	0	0	1	0	1	2
28	Pseudorca	0	3	1	0	2	1	1	0	1	1	1	1	1	0	1	1
29	Sotalia	0	2	1	2	2	2	1	1	1	2	1	1	1	0	1	1
30	Sousa	0	2	1	2	2	2	1	1	1	0	1	1	1	0	1	1
31	Stenella	0	2	1	2	2	2	1	1	1	0	1	1	1	0	1	1
32	Steno	0	1	1	2	2	2	1	1	1	1	1	1	1	0	1	1
33	Stenodelphis	1	2	1	1	1	1	1	1	1	0	1	1	1	0	1	1
34	Tasmacetus	0	3	1	2	2	0	0	0	2	0	1	1	2	0	*	2
35	Tursiops	0	2	1	2	2	2	1	1	1	0	1	1	1	0	1	1
36	Ziphius	0	1	1	2	2	0	0	0	2	1	1	1	2	1	2	2

The missing data are represented by "*".

Appendix A2. Classification of wild cats

Reference: Marcotorchino (1981). The parameters and entries in Table A2 are specified as follows:

(i) Morphological parameters.
1: *Aspect of the pelt.*
1: without spots, uniformly colored, 2: with spots, 3: with stripes, 4: marmoreal (like in rblé).
2: *Fur.*
0: short haired, 1: long haired.
3: *Ears.*
1: round or rounded, 2: pointed.
4: *Height (H) up to shoulder.*
1: $H \leq 50$ cm, 2: 50 cm $< H \leq 70$ cm, 3: $H > 70$ cm.
5: *Weight (W).*
1: $W \leq 10$ kg, 2: 10 kg $< W \leq 80$ kg, 3: $W > 80$ kg.
6: *Length (L) of body.*
1: $L \leq 80$ cm, 2: 80 cm $< L \leq 150$ cm, 3: $L > 150$ cm.
7: *Length of tail compared with length of body.*
1: short, 2: median, 3: long.
8: (*Teeth*) *Canines.*
0: little developed, 1: very developed.
9: (*Larynx*) *Lingual bone.*
0: absent, 1: present.
10: *Retractile claws.*
0: no, 1: yes.

(ii) Behavioral parameters.
11: *Predatory behavior.*
1: diurnal, 2: diurnal and nocturnal, 3: nocturnal.
12: *Type of prey.*
1: big prey (antelope, buffalo, etc.), 2: big or small prey, 3: small prey (shrewmouse, little monkey).
13: *Climbs trees.*
0: no, 1: yes.
14: *Chases after or lies in wait for the prey.*
0: wait, 1: chase.

Table A2
Data matrix and solution classes (SC)

	Wild cats	1	2	3	4	5	6	7	8	9	10	11	12	13	14	SC
1	Lion	1	0	1	3	3	3	2	1	1	1	1	1	0	1	1
2	Tigre	3	0	1	3	3	3	2	1	1	1	3	1	0	0	1
3	Jaguar	2	0	1	3	3	2	1	1	1	1	2	1	1	0	2
4	Leopard	2	0	1	3	3	2	2	1	1	1	3	2	1	0	2
5	Once	2	1	1	2	2	2	3	1	1	1	1	2	1	0	2
6	Guepard	2	0	1	3	2	2	3	0	0	0	1	2	0	1	3
7	Puma	1	0	1	2	3	2	3	1	0	1	2	2	1	0	2
8	Panth. Nebuleuse	2	0	1	2	2	2	3	1	1	1	3	3	1	0	2
9	Serval	2	0	2	2	2	2	1	0	0	1	1	3	1	1	4
10	Ocelot	2	0	1	2	2	2	2	0	0	1	2	3	1	0	4
11	Lynx	2	1	2	2	2	2	1	1	0	1	2	2	1	0	2
12	Caracal	1	0	2	2	2	1	1	0	0	1	2	3	1	1	4
13	C. Viverrin	2	0	1	1	1	2	2	0	0	1	2	3	0	0	4

Table A2—continued

	Wild cats	1	2	3	4	5	6	7	8	9	10	11	12	13	14	SC
14	Jaguarundi	1	0	1	1	2	2	1	0	0	1	*	1	1	0	4
15	C. Chaus	1	1	2	1	2	1	2	0	0	1	2	2	1	0	4
16	C. Dore	1	0	1	1	1	1	2	0	0	1	2	2	1	0	4
17	C. Marguay	2	0	1	1	1	1	2	0	0	1	2	2	1	0	4
18	C. Margerite	1	1	1	1	1	1	2	0	0	1	*	2	0	0	4
19	C. Cafer	3	0	1	1	1	1	2	0	0	1	2	2	1	1	4
20	C. Chine	1	0	2	1	1	1	1	0	0	1	*	2	1	0	4
21	C. Bengale	2	0	1	1	1	1	2	0	0	1	2	2	1	0	4
22	C. Rouilleux	2	0	1	1	1	1	2	0	0	1	2	2	1	0	4
23	C. Malais	1	1	1	1	1	1	1	0	0	1	2	2	1	0	4
24	C. Borneo	1	0	1	1	1	1	2	0	0	1	2	2	1	0	4
25	C. Negripes	2	0	1	1	1	1	1	0	0	1	2	2	1	1	4
26	C. Manul	1	1	1	1	1	1	1	0	0	1	2	2	1	0	4
27	C. Marbre	4	0	1	1	1	1	2	0	0	1	2	2	1	0	4
28	C. Tigrin	2	0	1	1	1	1	2	0	0	1	2	2	1	0	4
29	C. Temminck	1	0	1	1	1	1	2	0	0	1	2	2	1	0	4
30	C. Andes	2	1	1	1	1	2	2	0	0	1	2	*	1	0	4

Appendix A3. Classification of workers

Reference: Opitz and Schader (1984). The parameters in Table A3 are specified as follows:

1: *Type of work.* 2: *Working speed.* 3: *Noise level.* 4: *Humidity.* 5: *Friendly superiors.* 6: *Unfriendly superiors.* 7: *Headman.* 8: *Friendly colleagues.* 9: *Unfriendly colleagues.* 10: *Colleagues.* 11: *Salary.* 12: *Fairness w.r.t. payment.* 13: *Career perspectives.*

Table A3

Data matrix and solution classes (SC)

	1	2	3	4	5	6	7	8	9	10	11	12	13	SC
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	1	1	1	1	1	0	1	1	0	1	1	1	1
3	1	1	1	1	1	1	1	1	1	0	1	1	1	1
4	0	1	1	1	1	1	0	1	1	0	1	1	1	1
5	0	1	1	1	1	1	1	1	1	0	1	1	1	1
6	1	1	1	0	1	1	1	1	1	1	1	0	1	1
7	1	1	1	1	1	1	1	1	1	1	1	1	0	1
8	1	1	1	1	1	1	1	1	1	0	0	1	1	1
9	1	1	1	0	1	1	1	1	1	1	1	1	0	1
10	1	1	1	1	1	1	1	1	1	0	1	1	1	1
11	0	1	1	1	1	1	0	1	1	1	1	1	1	2
12	1	1	1	1	1	1	1	1	1	0	0	0	0	1
13	1	1	1	1	1	1	1	1	1	0	1	1	1	1
14	1	1	1	0	1	1	1	1	1	0	1	1	1	1
15	0	1	1	1	1	1	1	1	1	0	1	1	1	1
16	1	1	1	0	1	1	1	1	1	0	1	1	0	1
17	1	1	1	1	1	1	1	1	1	0	0	0	0	1
18	0	1	1	1	1	1	1	1	1	0	0	0	1	1
19	0	1	1	1	1	1	1	1	1	0	1	1	1	1

Table A3—continued

	1	2	3	4	5	6	7	8	9	10	11	12	13	SC
20	1	1	1	0	1	1	1	1	1	1	1	1	0	3
21	1	1	1	1	1	1	1	1	1	0	0	1	1	3
22	1	1	1	0	1	1	1	1	1	1	1	1	1	3
23	0	1	1	1	1	1	1	1	1	1	0	1	0	3
24	0	1	1	0	1	1	1	1	1	0	1	1	1	3
25	1	1	1	0	1	1	1	1	1	0	0	1	1	1
26	0	1	1	1	1	1	0	1	1	1	1	1	1	2
27	1	1	1	1	1	1	1	1	1	1	1	1	1	3
28	1	1	1	0	1	1	1	1	1	0	1	1	1	3
29	1	1	1	1	1	1	1	1	1	0	1	1	1	3
30	0	1	1	0	1	1	1	1	1	0	1	1	1	3
31	1	1	1	1	1	1	1	1	1	1	0	1	1	2
32	0	1	1	1	1	1	1	1	1	0	1	1	0	2
33	0	1	1	1	1	1	1	1	1	0	1	1	1	3
34	0	1	1	1	1	1	0	1	1	0	0	1	1	2

(11: Satisfied/many (5,6,8,9), 1 - 11: Unsatisfied/few, (0): Undecided.

Appendix A4. Classification of cars

Reference: Hartigan (1975). The parameters in Table A4 are specified as follows:

1: *Brake system.* 2: *Fuel system.* 3: *Electrical.* 4: *Exhaust.* 5: *Steering.* 6: *Engine, mechanical.* 7: *Rattles and squeaks.* 8: *Rear axle.* 9: *Rust.* 10: *Shock absorbers.* 11: *Transmission, clutch.* 12: *Wheel alignment.* 13: *Other.*

Table A4

Data matrix and solution classes (SC)

	Car	1	2	3	4	5	6	7	8	9	10	11	12	13	SC
1	AMC Ambassador 8	1	0	0	0	0	0	0	1	0	0	0	0	0	1
2	Buick Special 6	0	0	0	0	0	0	1	0	1	0	0	0	1	1
3	Buick Special 8	0	0	0	0	0	0	1	0	0	1	0	1	1	1
4	Buick 8 Full	0	0	0	1	0	1	1	0	1	1	0	1	0	1
5	Buick Riviera	0	0	1	1	0	0	0	0	0	1	0	0	0	1
6	Cadillac Chevy II	0	1	0	0	1	0	1	1	1	0	0	1	0	2
7	Chevelle 6	0	0	0	0	0	1	1	0	1	0	0	0	0	1
8	Chevelle 8	0	1	0	1	1	0	1	0	1	1	0	1	0	2
9	Chevrolet Full	0	1	1	1	1	0	1	1	1	1	1	1	0	2
10	Corsair 6	0	1	0	0	1	1	0	1	0	1	1	1	1	3
11	Corvette	0	0	0	1	0	0	1	1	0	0	1	0	0	1
12	Chrysler Newport	1	0	0	0	0	0	0	0	0	0	0	0	0	1
13	New Yorker	1	0	0	0	0	0	0	1	0	0	0	0	1	1
14	Dodge Full Size	1	0	0	0	0	0	1	0	0	1	0	0	0	1
15	Falcon 6	0	0	0	0	0	0	1	0	0	0	1	1	0	1
16	Fairlane 6	0	0	0	0	0	0	1	0	0	0	0	0	0	1
17	Fairlane 8	0	0	0	1	0	0	1	1	0	0	0	1	0	1
18	Ford, Full Size	0	0	0	1	1	0	0	0	0	1	0	1	1	1
19	Thunderbird	0	0	1	0	1	1	0	0	0	0	0	1	1	1

Table A4—continued

	Car	1	2	3	4	5	6	7	8	9	10	11	12	13	SC
20	Mercury Full	0	0	0	0	0	0	0	0	0	0	0	1	0	1
21	Olds Full	1	1	0	0	0	0	1	0	0	1	0	1	0	1
22	Plymouth Full	1	0	0	0	0	0	0	0	0	0	0	0	0	1
23	Pontiac Tempest	0	1	0	0	0	0	1	0	1	1	0	1	0	1
24	Pontiac Full	1	1	1	0	0	0	1	0	1	1	0	1	0	2
25	Rambler Rebel 6	0	0	1	0	0	0	0	1	0	0	1	0	1	1
26	Mercedes	0	0	0	0	0	0	0	0	0	1	0	0	0	1
27	MG 1100	0	0	1	1	0	0	0	0	0	0	0	0	0	1
28	Peugeot	0	0	0	0	0	0	0	0	0	0	1	0	0	1
29	Porsche	0	0	0	0	0	0	0	0	0	0	0	0	0	1
30	Renault	0	0	0	0	0	0	0	0	0	0	1	0	0	1
31	Volvo	0	0	0	1	0	0	0	1	0	0	0	0	0	1
32	VW bug	1	0	1	1	1	1	0	0	0	0	1	0	0	4
33	VW bus	0	0	1	0	0	1	0	0	1	0	1	0	0	1

The "1" means greater than average frequency of repair in 1963–1967

Appendix A5. Classification of micro computers

Reference: Chah (1984). The parameters in Table A5 are specified as follows.

1: *Color Monitor*, 2: *Disk Operating System CP/M*, 3: *Disk Operating System MS DOS*, 4: *Disk Operating System "other"*, 5: *Processor* (1: 8 bits, 2: 16 bits, 3: 32 bits), 6: *Parallel Interface*, 7: *Serial Interface*, 8: *IEEE 488 Interface*, 9: *Hard Disk* (0: if it does not exist, 1: 5 Mb, 2: 10 Mb), 10: *Number of diskette drives* (1 or 2), 11: *Price* (F.F.), 12: *Random Access Memory*, configuration (Kb), 13: *Random Access Memory*, maximum (Kb), 14: *Mass Storage*, Diskette Unit (Kb)

Table A5

Data matrix and solution classes (SC)

	Computer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	SC
1	PAP	0	0	2	0	2	1	1	1	0	1	70000	192	512	320	1
2	QX 10	1	2	0	0	1	2	1	1	0	2	135500	192	256	320	2
3	MACINTOSH	0	0	0	2	1	0	2	0	0	1	26000	128	512	400	1
4	TI PC	2	2	2	0	2	2	1	0	0	1	26000	128	768	320	1
5	PAP (2)	0	0	2	0	2	1	1	1	0	2	135000	192	512	320	1
6	APRICOT	0	0	2	0	2	2	2	1	0	2	98400	256	768	315	1
7	Z 150	0	0	2	0	2	2	2	0	0	2	58500	128	640	360	1
8	GOUPIL 3	0	2	0	0	1	2	2	0	0	2	59500	64	1024	360	2
9	APPLE 3	0	0	0	2	1	1	2	1	1	1	85000	256	256	140	5
10	TANDY 2000	1	0	2	0	2	2	2	0	0	2	80500	128	768	320	1
11	IBM PC	1	1	2	0	2	2	1	0	0	2	96100	128	640	320	1
12	TI PC (2)	2	2	2	0	2	2	1	0	0	2	89000	256	768	320	1
13	APPLE 2E	1	1	0	2	1	1	1	1	1	1	99400	128	832	140	5
14	TELE PC	1	0	2	0	2	2	2	1	2	1	89500	256	640	360	1
15	PAP (3)	0	0	2	0	2	1	1	1	2	1	87400	192	512	320	1
16	IBM PC XT	1	1	2	0	2	2	1	0	2	1	91000	128	640	320	1
17	Z 150 (2)	0	2	2	0	2	2	2	0	2	2	81500	128	640	360	1
18	TANDY 2000 (2)	1	0	2	0	2	2	2	0	2	2	82500	128	768	320	1

Table A5—continued

	Computer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	SC
19	VICTOR SI	1	2	2	0	2	2	2	1	2	2	66000	256	896	1228	4
20	T 200	0	2	0	0	1	2	2	0	0	2	22500	64	64	256	2
21	AS 100	2	1	2	0	2	2	2	2	0	2	32000	128	512	640	4
22	MZ 15	1	0	2	2	1	2	2	0	0	2	34000	136	372	400	4
23	BASIS 108	1	0	0	2	1	2	2	1	0	2	28500	384	384	160	3
24	LISA 2	0	0	0	2	3	2	1	1	0	1	35500	512	512	400	3
25	EUROPI PC	1	0	2	0	2	2	2	0	2	1	47400	128	1024	327	4
26	PSI 80	0	2	0	0	1	2	2	0	0	2	47800	80	256	308	2
27	CORONA PC 2	1	1	2	0	2	2	2	1	2	2	45000	256	512	320	4
28	OPLITE	1	0	2	0	2	2	2	0	0	2	33500	256	640	360	4
29	HORIZON	0	2	0	0	1	2	2	0	0	2	35000	64	576	360	2
30	FOXY	1	1	2	0	2	2	2	1	2	1	51000	256	1024	360	4
31	SKS 2500	0	2	0	0	1	1	2	1	0	2	32000	64	256	800	2
32	ZEPHYR	0	2	0	0	1	2	2	0	0	2	41400	64	64	640	2
33	MBC 4050	0	2	0	0	2	2	2	0	0	2	35600	256	1024	640	2
34	SANCO 8000	0	2	0	0	1	2	2	0	0	2	26100	70	192	400	2
35	IPC MODEL 1S	0	2	0	0	1	0	2	0	0	2	43000	64	512	782	2
36	DESKTOP 10	0	1	2	0	2	0	2	1	0	2	44800	128	768	360	4
37	LISA 2 S	0	0	0	2	3	2	1	1	1	1	47400	512	1024	400	3
38	NEC PC 8000	0	0	0	2	1	2	1	1	0	2	31800	32	64	320	3
39	M 20	0	0	0	2	2	2	2	1	0	2	21600	128	512	286	3
40	TRS 80 MOD 12	0	0	0	2	1	2	2	0	0	1	32000	80	768	422	2

(0): The feature does not exist, (1): The feature is optional, (2): The feature exists.

Appendix A6. Votes of 54 member states of the UNO

Reference: Marcotorchino (1981). See Table A6.

Table A6

Data matrix and solution classes (SC)

	States	a	b	c	SC		States	a	b	c	SC		States	a	b	c	SC
1	USA	0	0	0	1	16	COLO	5	2	1	3	31	LUXE	3	3	1	1
2	CANA	3	2	1	1	17	VENE	3	1	1	1	32	IRAN	3	2	3	4
3	CUBA	1	5	3	2	18	GUYA	5	1	1	3	33	SPAI	3	2	1	1
4	HAIT	5	1	1	3	19	ECUA	3	5	2	5	34	PORT	5	3	2	6
5	DOMI	1	1	1	2	20	PERU	3	1	1	1	35	POLA	1	1	3	2
6	JAMA	3	1	1	1	21	BRAZ	3	2	1	1	36	AUST	3	2	2	4
7	TRIN	3	1	1	1	22	BOLI	5	1	1	3	37	HUNG	1	1	3	2
8	BARB	5	1	2	3	23	PARA	3	2	1	1	38	CZEC	1	1	3	2
9	MEXI	3	1	1	1	24	CHIL	3	1	1	1	39	ITAL	3	2	1	1
10	GUAT	3	1	1	1	25	ARGE	3	1	1	1	40	MALT	5	5	1	3
11	HOND	3	2	1	1	26	URUG	3	5	1	1	41	ALBA	1	5	3	2
12	EL S	3	2	1	1	27	U.K.	3	3	1	1	42	YUGO	1	1	3	2
13	NICA	3	2	1	1	28	IREL	3	2	1	1	43	GREE	3	1	1	1
14	COST	3	1	1	1	29	NETH	3	3	1	1	44	CYPR	3	1	1	1
15	PANA	3	1	1	1	30	BELG	3	3	1	1	45	BULG	1	1	3	2

Table A6—continued

	States	a	b	c	SC
46	ROMA	1	1	3	2
47	USSR	1	1	3	2
48	UKRA	1	1	3	2
49	BYEL	1	1	3	2
50	FINL	2	2	3	4
51	SWED	3	2	3	4
52	NORW	3	2	3	4
53	DENM	3	2	3	4
54	ICEL	3	2	1	1

(1): In favour, (2): Against, (3): Abstaining, (5): Absent.

Appendix A7. Votes of all member states of the UNO

Reference: UNO (1985).

Remark. (a) The votes considered here are those which were recorded when the session was held.

(b) The states which announced that they were not participating in a vote are considered here as "absent".

A7.1. Votes on Resolutions 39/119-120-121

The parameters in Table A7.1 are specified as follows:

A: 39/119-Situation of human rights and fundamental freedoms in *El Salvador.*

B: 39/120-Situation of human rights and fundamental freedoms in *Guatemala.*

C: 39/121-Situation of human rights and fundamental freedoms in *Chile.*

The solution classes in Table A7.1 are specified as follows:

SC1: Solution classes with all states.

SC2: Solution classes without considering states which were absent at least once.

Table A7.1

Data matrix and solution classes

	States	A	B	C	SC1	SC2
1	Afganistan	0	0	0	1	1
2	Albania	0	3	3	2	
3	Algeria	0	0	0	1	1
4	Angola	0	0	0	1	1
5	Antigua and Barbuda	3	3	3	2	
6	Argentina	0	0	0	1	1
7	Australia	0	0	0	1	1
8	Austria	0	0	0	1	1
9	Bahamas	2	2	2	3	2
10	Bahrain	0	0	0	1	1
11	Bangladesh	1	1	1	4	3
12	Barbados	0	0	0	1	1
13	Belgium	0	0	0	1	1
14	Belize	2	2	2	3	2
15	Benin	0	0	0	1	1
16	Bhutan	2	2	2	3	2
17	Bolivia	3	3	3	2	
18	Botswana	0	0	0	1	1
19	Brazil	2	2	1	3	2
20	Brunei Darussalam	2	2	2	3	2
21	Bulgaria	0	0	0	1	1
22	Burkina Faso	0	0	0	1	1
23	Burma	2	2	2	3	2

Table A7.1—continued

	States	A	B	C	SC1	SC2
24	Burundi	0	0	0	1	1
25	Byelorussia	0	0	0	1	1
26	Cameroon	3	3	2	2	
27	Canada	0	0	0	1	1
28	Cape Verde	0	0	0	1	1
29	Central African Republic	2	2	2	3	2
30	Chad	2	2	2	3	2
31	Chile	1	1	1	4	3
32	China	2	2	2	3	2
33	Colombia	0	2	3	5	
34	Comoros	3	3	3	2	
35	Congo	0	0	0	1	1
36	Costa Rica	0	2	0	1	1
37	Cuba	0	0	0	1	1
38	Cyprus	0	0	0	1	1
39	Czechoslovakia	0	0	0	1	1
40	Democratic Kampuchea	2	2	2	3	2
41	Democratic Yemen	0	0	0	1	1
42	Denmark	0	0	0	1	1
43	Djibouti	3	3	3	2	
44	Dominica	3	3	3	2	
45	Dominican Republic	0	2	0	1	1
46	Ecuador	2	2	2	3	2
47	Egypt	0	2	2	3	2
48	El Salvador	1	1	1	4	3
49	Equatorial Guinea	2	2	0	3	2
50	Ethiopia	0	0	0	1	1
51	Federal Republic of Germany	2	0	0	1	1
52	Fiji	2	2	2	3	2
53	Finland	0	0	0	1	1
54	France	0	0	0	1	1
55	Gabon	2	2	2	3	2
56	Gambia	0	0	0	1	1
57	German Democratic Republic	0	0	0	1	1
58	Ghana	0	0	0	1	1
59	Greece	0	0	0	1	1
60	Grenada	3	3	3	2	
61	Guatemala	1	1	1	4	3
62	Guinea	0	2	0	1	1
63	Guinea-Bissau	3	3	3	2	
64	Guyana	0	0	0	1	1
65	Haiti	1	1	1	4	3
66	Honduras	1	2	2	3	2
67	Hungary	0	0	0	1	1
68	Iceland	0	0	0	1	1
69	India	0	0	0	1	1
70	Indonesia	1	1	1	4	3
71	Iran	0	0	0	1	1
72	Iraq	0	0	3	1	
73	Ireland	0	0	0	1	1
74	Israel	3	3	3	2	

Table A7.1—continued

	States	A	B	C	SC1	SC2
75	Italy	0	0	0	1	1
76	Ivory Coast	2	2	2	3	2
77	Jamaica	0	0	0	1	1
78	Japan	2	2	2	3	2
79	Jordan	2	2	2	3	2
80	Kenya	0	0	0	1	1
81	Kuwait	0	0	0	1	1
82	Lao People's Democratic Rep.	0	0	0	1	1
83	Lebanon	3	3	1	2	
84	Lesotho	0	0	0	1	1
85	Liberia	2	2	2	3	2
86	Libya	0	0	0	1	1
87	Luxembourg	0	0	0	1	1
88	Madagascar	0	0	0	1	1
89	Malawi	2	2	2	3	2
90	Malaysia	2	2	2	3	2
91	Maldives	2	2	0	3	2
92	Mali	0	0	0	1	1
93	Malta	0	0	0	1	1
94	Mauritania	0	0	0	1	1
95	Mauritius	0	0	0	1	1
96	Mexico	0	0	0	1	1
97	Mongolia	0	0	0	1	1
98	Morocco	1	1	1	4	3
99	Mozambique	0	0	0	1	1
100	Nepal	2	2	2	3	2
101	Netherlands	0	0	0	1	1
102	New Zealand	0	0	0	1	1
103	Nicaragua	0	0	0	1	1
104	Niger	2	2	2	3	2
105	Nigeria	0	2	2	3	2
106	Norway	0	0	0	1	1
107	Oman	2	2	2	3	2
108	Pakistan	2	1	1	4	3
109	Panama	0	2	2	3	2
110	Papua New Guinea	2	2	2	3	2
111	Paraguay	1	1	1	4	3
112	Peru	0	2	2	3	2
113	Philippines	2	2	2	3	2
114	Poland	0	0	0	1	1
115	Portugal	0	0	0	1	1
116	Qatar	0	0	0	1	1
117	Romania	2	2	0	3	2
118	Rwanda	0	0	0	1	1
119	Saint Lucia	3	3	3	2	
120	Saint Vicent	0	3	2	6	
121	Samoa	0	0	0	1	1
122	Sao Tome and Principe	0	0	0	1	1
123	Saudi Arabia	0	0	2	1	1
124	Senegal	0	0	0	1	1
125	Seychelles	0	0	0	1	1

Table A7.1—continued

	States	A	B	C	SC1	SC2
126	Sierra Leone	0	0	0	1	1
127	Singapore	2	2	2	3	2
128	Solomon Islands	3	3	3	2	
129	Somalia	2	2	2	3	2
130	Spain	0	0	0	1	1
131	Sri Lanka	2	2	0	3	2
132	St. Christopher and Nevis	3	3	3	2	
133	Sudan	2	2	2	3	2
134	Suriname	2	2	2	3	2
135	Swaziland	0	0	0	1	1
136	Sweden	0	0	0	1	1
137	Syria	0	0	3	1	
138	Thailand	2	2	2	3	2
139	Togo	0	0	0	1	1
140	Trinidad Tobago	2	2	2	3	2
141	Tunisia	0	0	0	1	1
142	Turkey	2	2	2	3	2
143	Uganda	0	0	0	1	1
144	Ukraine	0	0	0	1	1
145	USSR	0	0	0	1	1
146	United Arab Emirates	0	0	0	1	1
147	United Kingdom	2	0	0	1	1
148	United Republic of Tanzania	0	0	0	1	1
149	United States of America	1	1	1	4	3
150	Uruguay	1	1	1	4	3
151	Vanuatu	0	0	0	1	1
152	Venezuela	0	2	0	1	1
153	Viet Nam	0	0	0	1	1
154	Yemen	2	2	2	3	2
155	Yugoslavia	0	0	0	1	1
156	Zaire	2	2	2	3	2
157	Zambia	0	0	0	1	1
158	Zimbabwe	3	0	0	1	

A7.2. Votes on Resolution 39/148: "Review of the implementation of the recomendations and decisions adopted by the General Assembly at its tenth special session"

The parameters in Table A7.2 are specified as follows:

A: Unilateral nuclear disarmament measures. B: Bilateral nuclear arms negotiations. C: Nuclear weapons in all aspects. D: Non-use of nuclear weapons and prevention of nuclear war. E: Prohibition of the nuclear neutron weapon. F: Climatic effects of nuclear war: nuclear winter. G: Bilateral nuclear-arms negotiations. H: United Nations Institute for Disarmament Research. J: Disarmament Week. K: Cessation of the nuclear-arms race and nuclear disarmament. L: Implementation of the recomendations and decisions of the tenth special session. M: International co-operation for disarmament. N: Report of the Conference on Disarmament. O: Implementation of the recomendations and decisions of the tenth special session. P: Prevention of nuclear war.

The solution classes in Table A7.2 are specified as follows:

SC1: Solution class considering all states.

SC2: Solution class without considering states which were absent at least 8 times.

Table A7.2

Data matrix and solution classes

(*)	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	SC1	SC2
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
2	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
3	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
4	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
6	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	1	1
7	0	0	1	1	2	0	2	0	2	2	0	1	2	2	0	2	2
8	0	0	0	2	2	0	0	0	0	0	2	2	0	0	0	1	1
9	0	2	2	2	2	0	2	0	0	2	0	2	0	0	0	1	1
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
11	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	1
12	0	2	0	0	0	0	2	0	0	0	0	0	0	0	0	1	1
13	2	0	1	1	1	2	1	0	2	1	2	1	2	1	1	2	2
14	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
15	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
16	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	1	1
17	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	1
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
19	0	0	0	2	2	0	2	0	0	0	0	0	0	0	0	1	1
20	0	0	0	3	2	0	0	0	0	0	3	3	0	0	0	1	1
21	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	1	1
22	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
23	0	2	0	2	2	0	0	0	0	0	3	3	0	0	0	1	1
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
25	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	1	1
26	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	1
27	2	0	1	1	1	0	1	0	2	1	2	1	2	1	2	2	2
28	0	2	0	0	3	0	0	0	0	0	0	0	0	0	0	1	1
29	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
30	0	0	2	0	2	0	0	0	0	0	0	0	0	0	0	1	1
31	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	1	1
32	0	0	3	2	2	0	0	0	2	0	0	0	0	0	0	1	1
33	0	0	0	2	2	2	2	0	0	0	0	0	0	0	0	1	1
34	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
35	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
35	0	2	0	0	0	0	0	3	0	0	0	0	0	0	0	1	1
36	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	1	1
37	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
38	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
39	0	1	0	0	0	0	2	0	0	0	3	0	0	0	0	1	1
40	0	0	3	3	3	3	0	0	3	0	3	3	0	0	0	2	2
41	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
42	0	0	1	1	2	0	2	0	2	2	0	1	2	2	0	2	2
43	0	0	0	0	2	0	0	0	0	0	0	0	1	0	0	1	1
44	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	
45	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	1	1
46	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	1
47	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	1
48	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	1
49	0	0	3	0	2	0	0	0	0	0	0	0	0	0	0	1	1

Table A7.2 continued

(*)	A	B	C	D	E	F	G	H	J	K	L	M	N	O	P	SC1	SC2
50	0	2	0	0	0	0	0	0	0	0	2	0	0	0	0	1	1
51	2	0	1	1	1	2	1	0	2	1	2	1	2	1	1	3	2
52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
53	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	1
54	2	0	1	1	1	2	1	0	2	1	0	1	2	1	1	3	2
55	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	1
56	0	0	2	0	0	0	2	0	0	0	0	0	0	0	0	1	1
57	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	1	1
58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
59	0	2	0	0	2	0	0	0	0	0	0	2	2	0	0	1	1
60	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	
61	3	3	3	3	3	0	0	0	0	0	0	0	0	0	0	1	1
62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
63	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	
64	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
65	0	0	0	2	2	0	2	2	0	0	0	0	0	0	0	1	1
66	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	1	1
67	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	1	1
68	0	0	1	1	2	0	2	0	2	1	0	1	2	2	2	3	2
69	0	2	0	0	0	0	0	0	0	0	2	0	0	0	0	1	1
70	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	1	1
71	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	1	1
72	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	1	1
73	0	0	2	0	2	0	0	0	0	0	0	2	0	0	0	1	1
74	0	0	1	1	1	2	1	2	2	2	2	1	2	1	2	3	2
75	2	0	1	1	1	2	1	0	2	1	2	1	2	1	1	3	2
76	0	0	2	3	2	0	0	0	2	0	0	0	0	0	0	1	1
77	0	0	3	2	3	0	0	0	0	0	0	3	0	0	0	1	1
78	2	0	1	1	1	0	2	2	2	2	0	1	2	2	2	3	2
79	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
81	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	1	1
82	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
83	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	1	1
84	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
85	0	0	0	0	2	0	0	0	0	0	2	0	0	0	0	1	1
86	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
87	2	0	1	1	2	2	1	0	2	1	2	1	2	1	2	3	2
88	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
89	0	0	0	0	0	0	0	2	0	0	0	0	3	0	0	1	1
90	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
91	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	1
92	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
93	0	3	3	3	3	0	0	0	0	0	0	3	0	0	0	1	1
94	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
95	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
96	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
97	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	1	1
98	0	0	0	0	2	0	0	0	0	0	0	3	0	0	0	1	1
99	0	2	0	0	0	0	3	0	0	0	0	0	0	0	0	1	1
100	0	0	0	0	2	0	0	0	0	0	0	0	2	0	0	1	1

Table A7.2—continued

(*)	A	B	C	D	E	F	G	H	J	K	L	M	N	O	P	SC1	SC2
101	2	0	1	1	2	2	1	0	2	1	2	1	2	1	2	3	2
102	2	0	1	1	2	0	2	0	2	2	2	1	2	2	2	3	2
103	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
104	0	0	2	0	0	0	2	0	0	0	0	0	0	0	0	1	1
105	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	1	1
106	0	0	1	1	2	0	2	0	2	1	0	1	2	2	2	3	2
107	0	0	0	0	2	0	0	0	3	0	0	0	0	0	0	1	1
108	0	0	0	0	2	0	0	0	0	0	2	0	0	0	0	1	1
109	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
110	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	
111	0	0	2	2	2	0	0	0	2	2	0	2	0	0	0	1	1
112	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	1
113	0	0	3	2	2	0	2	0	0	0	0	3	0	0	0	1	1
114	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	1	1
115	2	0	1	1	1	0	1	0	0	1	0	1	2	1	2	3	2
116	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
117	0	3	0	0	0	0	3	0	0	0	0	0	0	0	0	1	1
118	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	1	1
119	0	0	0	2	2	0	0	0	0	2	0	0	0	0	0	1	1
120	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	1	1
121	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	
122	0	2	3	0	3	0	3	0	0	0	0	0	0	0	0	1	1
123	3	3	3	3	3	3	3	0	0	0	0	0	0	0	0	1	1
124	0	0	2	0	2	0	0	0	0	0	0	3	0	0	0	1	1
125	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	1	1
126	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
127	0	0	0	3	2	0	0	0	3	0	3	3	0	0	0	1	1
128	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	
129	3	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	1
130	0	0	1	1	2	0	2	0	0	2	0	1	2	2	2	3	2
131	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	1
132	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	
133	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1	1
134	0	2	0	0	0	3	0	0	0	0	0	0	0	0	0	1	1
135	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	
136	0	0	2	0	2	0	0	0	0	0	2	2	0	0	0	1	1
137	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
138	0	0	0	0	2	0	0	0	0	0	3	0	0	0	0	1	1
139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
140	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
141	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
142	2	0	1	1	1	2	1	0	2	1	0	1	2	1	2	3	2
143	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
144	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	1	1
145	0	1	0	0	0	0	2	0	0	0	0	0	0	0	0	1	1
146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
147	2	0	1	1	1	2	1	0	2	1	2	1	2	2	1	3	2
148	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
149	1	0	1	1	1	2	1	1	2	1	2	1	1	1	1	3	2
150	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	1	1
151	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	2	

Table A7.2—continued

(*)	A	B	C	D	E	F	G	H	J	K	L	M	N	O	P	SC1	SC2
152	0	2	0	0	2	0	0	0	0	0	0	0	0	0	0	1	1
153	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
154	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
155	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
156	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	1	1
157	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
158	3	3	3	0	0	0	0	0	0	0	0	0	0	0	0	1	1

(0): In favour, (1): Against, (2): Abstaining, (3): Absent.
(*) The States 1 to 158 are precisely those given in Table A7.1.

A7.3. Votes on Resolution 39/99: United Nations Relief and Works for Palestine Refugees in the Near East

The parameters are specified as follows:

A: Assistance to Palestine refugees. D: Offers by Member States of grants and scholarships for higher education including vocational training, for the Palestine refugees. E: Palestine refugees in the Gaza Strip. F: Resumption of the ration distribution to Palestine refugees. G: Population and refugees displaced since 1967. H: Revenues derived from Palestine refugee properties. I: Protection of Palestine refugees. J: Palestine refugees in the West Bank. K: University of Jerusalem "Al-Quds" for Palestine refugees.

For the data matrix see UNO (1985) or Wakabayashi (1986). Consider the states from 1 to 158 as in Table A7.1 and the 9 motions above specified. The solution classes are specified as follows:

Solution classes considering all states:
Class 1: 5, 34, 44, 60, 71, 119, 120, 128, 132, 135, 151. Class 2: 74, 149. Class 3: all the others.
Solution classes without considering states which were absent at least 5 times:
Class 1: 74, 149. Class 2: all the others.

Appendix A8. Classification of companies

Reference: Späth (1977). The complete data set can be found in Späth (1977). The solutions classes are specified as follows:

Class 1: 10 and 107. Class 2: all the others.

Acknowledgement

The authors would like to thank O. Opitz (Augsburg) for introducing them to qualitative data analysis and suggesting the research project on which this paper reports.

References

F. Barahona, M. Grötschel, M. Jünger and G. Reinelt, "An application of combinatorial optimization to statistical physics and circuit layout design," *Operations Research* 36 (1988) 493–513.

J.P. Barthélemy and B. Monjardet, "The median procedure in cluster analysis and social choice theory," *Mathematical Social Sciences* 1 (1981) 235-267.

J.A. Bondy and U.S.R. Murty, *Graph Theory with Applications* (Macmillan, London, 1976).

S. Chah, "Classification of heterogeneous data: micro computers," Paper presented at the III International Symposium on Data Analysis (Brussels, 1985).

H.P. Crowder and M.W. Padberg, "Solving large scale travelling salesman problems to optimality," *Management Science* 26 (1980) 495-509.

M. Grötschel, L. Lovász and A. Schrijver, "The ellipsoid method and its consequences in combinatorial optimization", *Combinatorica* 1 (1981) 169-197.

M. Grötschel, M. Jünger and G. Reinelt, "A cutting plane algorithm for the linear ordering problem", *Operations Research* 32 (1984) 1195-1220.

M. Grötschel and O. Holland, "Solving matching problems with linear programming," *Mathematical Programming* 33 (1985) 243-259.

M. Grötschel and Y. Wakabayashi, "Facets of the clique partitioning polytope", Report No. 6, Schwerpunktprogramm der Deutschen Forschungsgemeinschaft Universität Augsburg (Augsburg, West Germany, 1987), to appear in *Mathematical Programming*.

J.A. Hartigan, *Clustering Algorithms* (Wiley, New York, 1975).

J.F. Marcotorchino, *Agrégation des Similarités en Classification Automatique*, These de Doctorat d'état (Université Paris vi, 1981).

J.F. Marcotorchino and P. Michaud, "Optimisation en analyse des données relationnelles," in: E. Diday, et al. (eds.), *Data Analysis and Informatics* (North-Holland, Amsterdam, 1980) pp. 655-670.

J.F. Marcotorchino and P. Michaud, "Optimization in exploratory data analysis," Proceedings of 5th International Symposium on Operations Research 1981 (Physica Verlag, Köln, 1981a).

J.F. Marcotorchino and P. Michaud, "Heuristic approach to the similarity aggregation problem," *Methods of Operations Research* 43 (1981b) 395-404.

O. Opitz and M. Schader, "Analyse qualitativer Daten: Einführung und Übersicht," *Operations Research Spektrum* 6 (1984) 67-83.

M. Padberg and G. Rinaldi, "Optimization of a 532-city symmetric traveling salesman problem by branch and cut," *Operations Research Letters* 6 (1987) 1-7.

G. Reinelt, *The Linear Ordering Problem: Algorithms and Applications* (Research and Exposition in Mathematics 8) (Helderman Verlag, Berlin, 1985).

M. Schader and U. Tüshaus, "Ein Subgradientenverfahren zur Klassifikation qualitativer Daten," *Operations Research Spektrum* 7 (1985) 1-5.

A. Schrijver, *Theory of Linear and Integer Programming* (Wiley, Chichester, UK, 1986).

H. Späth, "Partitionierende Cluster-Analyse für große Objektmengen mit binären Merkmalen am Beispiel von Firmen und deren Berufsgruppenbedarf," in: H. Späth, (ed.), *Fallstudien Cluster Analyse* (Oldenburg, München, 1977) pp. 63-80.

U. Tüshaus, "Aggregation binärer Relationen in der qualitativen Datenanalyse," in: *Mathematical Systems in Economics Vol. 82* (Hain, Königstein, 1983).

UNO, "Resolutions and Decisions adopted by the General Assembly during the first part of its thirty-ninth Session" (from Sept. 18 to Dec. 18, 1984), (1985) 412-419.

G. Vescia, (a) "Descriptive classification of cetacea: whales, porpoises and dolphins," (b) "Automatic classification of cetaceans by similarity aggregation," in: J.F. Marcotorchino, J.M. Proth and J. Janssen, (eds.), *Data Analysis in Real Life Environment: Ins and Outs of Solving Problems* (Elsevier Science Publishers B.V. (North-Holland, Amsterdam, 1985) pp. 7-24.

Y. Wakabayashi, *Aggregation of Binary Relations: Algorithmic and Polyhedral Investigations*, Ph.D. Thesis (Universität Augsburg, West Germany, 1986).

Mathematical Programming 45 (1989) 97-108
North-Holland

A GENERALIZATION OF ANTIWEBS TO INDEPENDENCE SYSTEMS AND THEIR CANONICAL FACETS

Monique LAURENT
CNET PAA-TIM, 38-40 rue du général Leclerc, 92131 Issy-les-Moulineaux, France

Received 20 January 1987
Revised manuscript received 25 February 1988

We consider independence system polytopes, i.e. polytopes whose extreme points are the incidence vectors of the sets of an independence system. We first give a sufficient condition for recognizing Boolean facets. Then, the notion of antiweb introduced by Trotter for graphs is generalized to independence systems and used for obtaining canonical facets of the associated polytopes. We also point out how our results relate with known ones for knapsack, set covering and matroid polytopes.

Key words: 0, 1 integer programming, independence system, facet, antiweb.

1. Introduction

Given a finite set $E = \{e_1, \ldots, e_n\}$, an *independence system* (IS for short) on E is a family $\mathcal{J}$ of subsets of E closed under inclusion, i.e. satisfying the following property:

(I1) If $J \in \mathcal{J}$ and $I \subseteq J$, then $I \in \mathcal{J}$.

A set belonging to $\mathcal{J}$ is called *independent* and a set that does not belong to $\mathcal{J}$ is called *dependent*. Minimal (for set inclusion) dependent sets are called *circuits* and we denote by $\mathcal{C}(\mathcal{J})$ the family of circuits of $\mathcal{J}$. The collection $\mathcal{C}(\mathcal{J})$ is a *clutter*, i.e., if $C, C' \in \mathcal{C}(\mathcal{J})$ and $C \subseteq C'$, then $C = C'$. An independence system is fully characterized by its family of circuits and, conversely, every clutter $\mathcal{C}$ determines a unique IS: $\mathcal{J}(\mathcal{C}) = \{I \subseteq E: C \not\subset I \text{ for all } C \in \mathcal{C}\}$. The *rank function* of the IS $\mathcal{J}$ is the set function defined by $r(S) = \max(|I|: I \in \mathcal{J} \text{ and } I \subseteq S)$ for all $S \subseteq E$. We also define the *independence number* $\alpha(\mathcal{J})$ of the IS $\mathcal{J}$ as the maximum size of the independent sets, i.e. $\alpha(\mathcal{J}) = r(E)$. Notice that, if all circuits have size two, then $\mathcal{J}$ is the family of stable sets of the graph G with E as nodeset and $\mathcal{C}$ as edgeset and the independence number is exactly the stability number of G as defined in [1]. For a subset $S \subseteq E$, the family $\mathcal{J}_S = \{I \in \mathcal{J}: I \subseteq S\}$ is clearly an IS on S whose family of circuits is given by $\mathcal{C}_S = \{C \in \mathcal{C}: C \subseteq S\}$ and whose independence number satisfies $\alpha(\mathcal{J}_S) = r(S)$ (note that, when $\mathcal{J}$ is a matroid, this is the classical notion of restriction, cf. [17, Chap. 4]).

Given an IS $\mathcal{J}$ on E, we define the *independence system polytope*: $P = P(\mathcal{J}) = \text{Conv}(\mathcal{J})$ of $\mathbb{R}^E$ to be the convex hull of the incidence vectors of the independent sets of $\mathcal{J}$. In many applications, a weight w_e is associated with each element $e \in E$