RZG RECHENZENTRUM GRAZ

The Solution Sets of Extremal Equations

104

The Solution Sets of Extremal Equations

Günter Rote

Technische Universität Graz
Institut für Mathematik
Kopernikusgasse 24, A-8010 Graz

December 1984

Abstract:

Systems of linear extremal equations over linearly ordered sets are considered. These systems include "maxsum" and "maxmin" equation systems. Their solution sets will be characterized in several ways. The combinatorial structure of these characterizations can be described by a generalized set covering problem. We give algorithms for enumerating all minimal solutions of these generalized set covering problems.

Diese Arbeit wurde durch das Rechenzentrum Graz und den Fonds zur Förderung der wissenschaftlichen Forschung, Projekt S 32/01, gefördert.

1. Introduction

Systems of extremal equations, i. e. equations involving max or min operations, have found considerable interest in the literature, both theoretically and from practical viewpoints (cf. the examples below and the references given there; see also the monograph by U. Zimmermann [1981]).

In this paper systems of equations of the following form shall be considered:

Let $B_1, B_2, \ldots, B_m$, and $X_1, X_2, \ldots, X_n$ be linearly ordered sets; Given $b_i \in B_i$ $(i=1,2,\ldots,m)$ and isotone functions $f_{ij}: X_j \to B_i$ $(i=1,2,\ldots,m;$ $j=1,2,\ldots,n)$, find the set X of all $\underline{x}=(x_1,x_2,\ldots,x_n) \in X_1 \times X_2 \times \ldots \times X_n$, for which

$$(1) \qquad \max_{1 \leq j \leq n} f_{ij}(x_j) = b_i, \quad \text{for } i=1,2,\ldots,m,$$

This linear system of extremal equations can be interpreted as a vector equation $\underline{F} \cdot \underline{x} = \underline{b}$, where in the matrix-vector product $\underline{F} \cdot \underline{x}$ multiplication is replaced by applying a function to an argument and addition is replaced by the max operation.

In the examples considered we have $B_1=\ldots=B_m=X_1=\ldots=X_n=:B$ and $f_{ij}(x) = a_{ij} \circledast x$ for some isotone algebraic operation $\circledast: B\times B \rightarrow B$ and given elements $a_{ij} \in B$. Then the system (1) takes the form

$$(1') \qquad \max_{1\leq j\leq n} a_{ij} \circledast x_j = b_i, \quad \text{for } i=1,2,\ldots,m.$$

With different algebraic structures $(B,\circledast)$ one obtains different specializations of the original problem:

(a) $(B,\circledast)$ is a linarly ordered group, for example the group $(\mathbb{R},+)$. Applications include for example machine-scheduling problems. Such systems have been studied extensively by Cuninghame-Green [1979, chapters 14 and 15] and K. Zimmermann [1976] and have previously been dealt with by Vorob'ev [1963, 1967], K. Zimmermann [e. g. 1973-a, 1973-b, 1974-a, 1974-b] and U. Zimmermann [1979]

(b) $\circledast$ is the min-operation. Such systems arise in connection with fuzzy relation equations on fuzzy sets [Czogała, Drewniak, and Pedrycz 1982], [Sanchez 1976].

(c) $B=\{0,1\}$ and $\circledast$ is the min-operation (equations in the two-element Boolean algebra). This is a special case of (b). In this case the system (1') describes the feasible solutions of the set covering problem, which is a fundamental combinatorial problem with many applications (see e. g. [Garfinkel and Nemhauser 1972, especially chapter 8: the set covering and partitioning problems]). An overview of results concerning general systems of equations in Boolean algebras can be found in [Hammer and Rudeanu 1968, chapter II].

Most of the cited literature (cf. also K. Zimmermann [1982, 1984]) deals with optimization problems over the set of feasible solutions described by a system of type (1). This paper is restricted to describing the entire set of solutions of the system (1). Several elementary characterizations of the solution set will be derived in section 2. It turns out that essentially the solution set has one maximal element but in general several minimal elements. The set of minimal elements will be characterized combinatorically as corresponding to the minimal solutions of some sort of generalized set covering problem. For the special case of the group $(\mathbb{R}^+,\cdot)$ of positive real numbers (cf. section 4), the connection between extremal equation systems and set coverings has been known for a long time (cf. Vorob'ev [1963, 1967]; K. Zimmermann [1973-b, 1974-a, 1974-b] used this connection algorithmically to solve optimization problems for which the set of feasible solutions is described by systems of extremal equations and inequalities). K. Zimmermann [1976] and K. Zimmermann and Juhnke [1979] considered systems of extremal equations and inequalities over linearly ordered monoids. The latter paper is entirely devoted to characterizing the set of vectors defined by a system of extremal equations or inequalities. Section 3 contains two algorithms for enumerating all these minimal solutions. These algorithms are not fast for the general case, and they are included only for the sake of completeness and since they are later specialized for the above examples (a)-(c) in sections 4-6. According to Lawler, Lenstra, and Rinnooy Kan [1980], no algorithm for enumerating the minimal set coverings the running time of which is polynomial in the problem size and in the number of minimal coverings is known, nor has it been proved that the existence of such an algorithm would imply P=NP. Hence, for the time being, it seems justified to write such exponential algorithms.

For the second example (maxmin equations) the given algorithm is very efficient unless the problem degenerates too much (i. e. if many of the b_i values on the right-hand side are equal). For maxmin equations, bounds on the number of minimal elements are derived, which seem to be quite tight.

2. The set of solutions of the system (1)

By the isotonicity of f_{ij}, the solution set I_{ij} of the equation

$$f_{ij}(x) = b_i,$$

is a convex subset of X_j and may be represented as the difference of two initial segments of X_j:

$$I_{ij} = u_{ij} \setminus l_{ij},$$

with $u_{ij} = \{x \in X_j \mid f_{ij}(x) \leqq b_i\}$, and

$l_{ij} = \{x \in X_j \mid f_{ij}(x) < b_i\}$.

l_{ij} and u_{ij} could be replaced by the greatest lower bound or the least upper bound of I_{ij}, respectively, if these were known to exist. It might help the reader to assume that the infima and suprema exist and to read the phrases $x \in u_{ij}$, $x \notin u_{ij}$, $x \in l_{ij}$, $x \notin l_{ij}$, which shall occur frequently in the sequel, as denoting $x \leqq u_{ij}$ or $x < u_{ij}$, $x > u_{ij}$ or $x \geqq u_{ij}$, $x < l_{ij}$ or $x \leqq l_{ij}$, $x \geqq l_{ij}$ or $x > l_{ij}$, as appropriate, depending on whether the supremum or infimum is attained or not, and to read $l_{ij} \subset u_{ij}$ and $u_{ij} \subseteq l_{ij}$ as $l_{ij} \leqq u_{ij}$ and $l_{ij} > u_{ij}$ or $l_{ij} < u_{ij}$ and $l_{ij} \geqq u_{ij}$, as appropriate.
(The Symbol $\subseteq$ denotes set inclusion, and $\subset$ denotes proper set inclusion. $\not\subseteq$ is the negation of $\subseteq$. For initial segments a and b like the sets u_{ij} and l_{ij}, $a \not\subseteq b$ is equivalent to $b \subset a$.)

Now, for all $i=1,2,\dots,m$:

$$\max_{1\leq j\leq n} f_{ij}(x_j) = b_i \iff \max_{1\leq j\leq n} f_{ij}(x_j) \leq b_i \text{ and } \max_{1\leq j\leq n} f_{ij}(x_j) \geq b_i .$$

Since

$$\max_{1\leq j\leq n} f_{ij}(x_j) \leq b_i \iff$$

$$\text{for all } 1\leq j\leq n:\ f_{ij}(x_j) \leq b_i \iff$$

$$\text{for all } 1\leq j\leq n:\ x_j \in u_{ij},$$

and since

$$\max_{1\leq j\leq n} f_{ij}(x_j) \geq b_i \iff$$

$$\text{for some } 1\leq j\leq n:\ f_{ij}(x_j) \geq b_i \iff$$

$$\text{for some } 1\leq j\leq n:\ x_j \notin l_{ij},$$

we get the following characterization of the solution set X of (1):

(2) $$X = \{ (x_1,\dots,x_n) \mid \text{for all } 1\leq j\leq n:\ x_j \in u_j;\ \text{and for all } 1\leq i\leq m \text{ there is some } 1\leq j\leq n \text{ such that } x_j \notin l_{ij} \},$$

where $u_j := \bigcap_{1\leq i\leq m} u_{ij}$.

The first condition, "$x_j \in u_j$, for all j", is easy to check; to visualize the second condition, let us construct an m×n 0-1-matrix $R(\underline{x}) = (r_{ij}(\underline{x}))$, where $r_{ij}(\underline{x})=1$ if and only if $x_j \notin l_{ij}$. The second condition states then that each row of this matrix must contain at least one 1. For this situation we shall also say that the sets $P^j(\underline{x})=\{i|r_{ij}(\underline{x})=1\}$ corresponding to the columns $j=1,2,\ldots,n$ form a <u>covering</u> of the set of rows $\{1,2,\ldots,m\}$. Note that the j^{th} column of the matrix $R(\underline{x})$ and with it the set $P^j(\underline{x})$ depends only on the component x_j of $\underline{x}$.

For a given problem the columns of the matrix $R(\underline{x})$ cannot be arbitrary 0-1 columns, since $r_{i_1 j}(\underline{x})=1$ implies $r_{i_2 j}(\underline{x})=1$ if $l_{i_1 j} \supseteq l_{i_2 j}$. Rather, we have at most m+1 possibilities for each column of the matrix, ranging from the zero column (unless some $l_{ij}=\emptyset$), to the column containing only ones. As x_j increases, the set $P^j(\underline{x})$ increases with it stepwise at certain points, either by one element at a time or by several elements at a time, if the corresponding sets l_{ij} are equal.

Given a matrix R_0 which fulfills the above condition that $r_{i_1 j}=1$ implies $r_{i_2 j}=1$ if $l_{i_1 j} \supseteq l_{i_2 j}$, in all columns j, it is easy to see that (with two exceptions) an element $\underline{x}$ can be constructed such that $R(\underline{x})=R_0$: x_j must be contained in all sets l_{ij} with $r_{ij}=0$ but in none of the sets l_{ij} with $r_{ij}=1$. The condition ensures that there is such an x_j, unless column j contains only zeros or only ones. In the first case it is necessary in addition that no l_{ij} in the j^{th} column is empty, in the second case it is required that no l_{ij} is equal to the whole set X_j.

If $u_j \subseteq l_{ij}$ for some pair (i,j) then it is clear that for a vector $\underline{x} \in X$, $r_{ij}(\underline{x})$ cannot be 1. In other words, the "1" elements of the matrix $R(\underline{x})$ are a subset of the "1" elements of the matrix $R=(r_{ij})$ where

$$r_{ij}=1 \iff l_{ij} \subset u_j \iff u_j \not\subseteq l_{ij}.$$

(Shortly, we will say that the 0-1-matrix $R(\underline{x})$ is included in the 0-1-matrix R, or that R is greater that $R(\underline{x})$, $R \geqq R(\underline{x})$.)

Deciding whether the system (1) is solvable

From the above remarks it follows that there can be no solution of (1) if R contains a zero row. Furthermore, if $u_j=\emptyset$ for some j, then X is empty as well. On the other hand, if neither of these conditions hold, then, like in the case that has been discussed, an $\underline{x}$ can be constructed such that $R(\underline{x})=R$ and $x_j \in u_j$ for all j (see the example below).
Thus, the existence of a solution of (1) can be decided by checking n relations $u_j \neq \emptyset$ and mn relations $l_{ij} \subset u_j$.

We define $U := \{(x_1,\ldots,x_n) \mid \text{for all } 1 \leq j \leq n: x_j \in u_j\} = u_1 \times u_2 \times \ldots \times u_n$. Thus U is the set of solutions of the system

$$\max_{1 \leq j \leq n} f_{ij}(x_j) \leq b_i, \quad \text{for } i=1,2,\ldots,m.$$

Let $L_i := \{(x_1,\ldots,x_n) \mid \text{for all } 1 \leq j \leq n: x_j \in l_{ij}\} = l_{i1} \times l_{i2} \times \ldots \times l_{in}$, for all $1 \leq i \leq m$; then we get

$$(x_1,\ldots,x_n) \in L_i \iff \text{there is no } j \text{ such that } f_{ij}(x_j) \geq b_i,$$
$$\iff \max_{1 \leq j \leq n} f_{ij}(x_j) < b_i.$$

Thus L_i contains exactly those elements of U which violate the i^{th} equation of the system (1), and possibly additional elements which are not contained in U. Hence we get

Characterization 1:

$$X = U \setminus \bigcup_{1 \leq i \leq m} L_i .$$

The set $X_1 x X_2 x \ldots x X_n$ with the partial order which is the product of the linear orders on the sets X_j is a lattice; all of the m+1 sets L_i and U in the theorem are ideals in this lattice. They are cartesian products of initial segments of the sets X_j.

Example 1:

Let n=2, $X_1=X_2=\mathbb{R}^+ \cup \{0\}$ and consider the following set of equations:

$$\max(\min(x_1-1,0), \min(x_2-5,0)) = 0$$
$$\max(\min(x_1-3,0), \min(x_2-3,0)) = 0$$
$$\max(\min(x_1-2,0), \min(x_2-1,0)) = 0$$
$$\max(\max(x_1-5,0), x_2-7) = 0$$

We obtain the following values for l_{ij}, u_{ij}, and u_j:

i: | j:	l_{i1}	u_{i1}	l_{i2}	u_{i2}
1	[0,1)	$\mathbb{R}^+$	[0,5)	$\mathbb{R}^+$
2	[0,3)	$\mathbb{R}^+$	[0,4)	$\mathbb{R}^+$
3	[0,2)	$\mathbb{R}^+$	[0,1)	$\mathbb{R}^+$
4	$\emptyset$	[0,5]	[0,7)	[0,7]
u_j:		[0,5]		[0,7]

L_4 is empty. The sets U, L_1, L_2, L_3, and X are shown in figure 1.

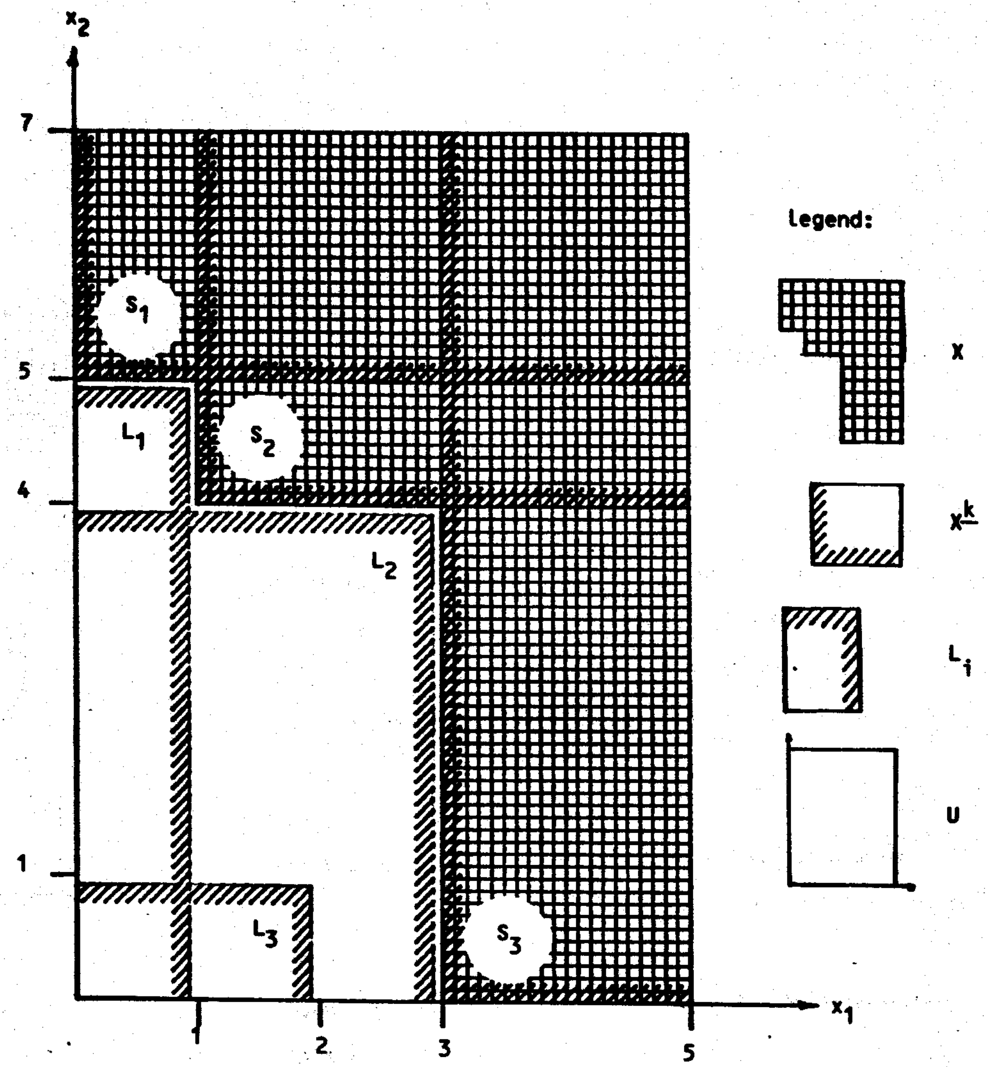

Figure 1 - the sets X, L_1, L_2, and L_3 for example 1; $S_1=X^{(2,2,2,1)}$, $S_2=X^{(1,2,2,1)}$, and $S_3=X^{(1,1,1,1)}$; the set U is the big rectangle.

The combinatorial structure of the problem (for example the fact that $L_3 \subsetneq L_2$ but L_1 and L_2 are incomparable, or that there are three minimal solutions) would not change if for example $l_{22}=[0,4)$ were replaced by $l_{22}=[0,3)$. Only the order among the l_{i2} is relevant. To emphasize these combinatorial properties of the problem the matrices $R(\underline{x})$ have been introduced above. For example, for the points $\underline{x}=(2,4)\epsilon U$ and $\underline{x}=(2,2)\epsilon U$ we get

$$R(2,4) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}, \quad R(2,2) = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix}.$$

Thus $(2,4)\epsilon X$, but $(2,2)\not\epsilon X$. The matrix R defined above is the full matrix

$$R = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

To construct an $\underline{x}\epsilon U$ sucht that $R(\underline{x})=R$, as was asserted to be possible if $U\neq\emptyset$, x_1 must be an element of u_1 but it must not be contained in l_{i1} for all i for which $r_{i1}=1$, that is, for all $i=1,2,3,4$. We get $x_1\epsilon[3,5]$. Similarly, we get $x_2=7$. Thus, for all $\underline{x}\epsilon[3,5]\times\{7\}$, $R(\underline{x})=R$.

A second representation of X can be derived from (2):
We can rewrite (2) as follows:

$$X = \{ (x_1,\dots,x_n) \mid \text{for all } 1\leq j\leq n\text{: } x_j \in u_j\text{; and for all } 1\leq i\leq m \text{ there is some } k(i),\ 1\leq k(i)\leq n \text{ such that } x_{k_i} \notin l_{ik_i} \} =$$

$$= \{ (x_1,\dots,x_n) \mid \text{for all } 1\leq j\leq n\text{: } x_j \in u_j\text{; and there is some sequence } \underline{k}=(k_1,k_2,\dots,k_m) \in \{1,\dots,n\}^m\text{, such that for all } 1\leq i\leq m \ \ x_{k_i} \notin l_{ik_i} \}.$$

Let $\underline{k} = (k_1,k_2,\ldots,k_m) \in \{1,\ldots,n\}^m$.

We define

$$(3) \qquad X^{\underline{k}} := \{(x_1,\ldots,x_n) \mid \begin{array}{l} \text{for all } 1\leq j\leq n\text{: } x_j \in u_j\text{; and} \\ \text{for all } 1\leq i\leq m\text{: } x_{k_i} \notin l_{ik_i}\}. \end{array}$$

In other words, $k_i=j$ states that for every $\underline{x}\in X^{\underline{k}}$ the i^{th} row of the matrix $R(\underline{x})$ will contain a 1 in the place $r_{ij}(\underline{x})$.

The following characterization is now an immediate consequence:

<u>Characterization 2</u>:

$$X = \bigcup_{\underline{k}\in\{1,\ldots,n\}^m} X^{\underline{k}}.$$

Notational remark:

The names i,p,q, possibly subscripted, shall be reserved for row indices in the range [1,m] or [0,m], whereas the names j,k, and of course also k_i,j', etc., shall denote column indices in the range [1,n] (and sometimes [0,n]). Likewise $\underline{p}$,$\underline{k}$, etc. denote vectors of indices of the respective types, P,K, etc. denote sets of such indices, and $\underline{P}$,$\underline{K}$', etc. denote sets of vectors of such indices.

From the definition (3) it follows that $X^{\underline{k}}$ can also be written as:

$$X^{\underline{k}} = \{ (x_1,\ldots,x_n) \mid \text{for all } 1\leq j\leq n\text{: } x_j \in u_j \setminus L^{\underline{k}}_j \} =$$
$$= (u_1 \setminus L^{\underline{k}}_1) \times \cdots \times (u_n \setminus L^{\underline{k}}_n),$$

where $L^{\underline{k}}_j = \bigcup_{k_i=j} l_{ik_i} = \bigcup_{k_i=j} l_{ij}$.

Such a set $L^{\underline{k}}_j$ is equal to some l_{ij} (namely the largest one with $k_i=j$) or empty.

Thus each $X^{\underline{k}}$ has the form

(4) $$X_{\underline{p}} := (u_1 \setminus l_{p_1 1}) \times \cdots \times (u_n \setminus l_{p_n n}),$$

for some sequence $\underline{p}=(p_1,\ldots,p_n)$ with $0 \leq p_j \leq m$, if we set $l_{0j}=\emptyset$ for all j. $\underline{p}$ is called the sequence generated by $\underline{k}$.

This is the point where the isotonicity of the functions f_{ij} is crucial.

Example

With the data of example 1 above, we obtain

$X^{(1,1,1,1)}=X_{(2,0)}$ = [3,5]×[0,7] $X^{(2,1,1,1)}=X_{(2,1)}$ = [3,5]×[5,7]
$X^{(1,1,1,2)}=X_{(2,4)}$ = [3,5]×{7} $X^{(2,1,1,2)}=X_{(2,4)}$ = [3,5]×{7}
$X^{(1,1,2,1)}=X_{(2,3)}$ = [3,5]×[1,7] $X^{(2,1,2,1)}=X_{(2,1)}$ = [3,5]×[5,7]
$X^{(1,1,2,2)}=X_{(2,4)}$ = [3,5]×{7} $X^{(2,1,2,2)}=X_{(2,4)}$ = [3,5]×{7}
$X^{(1,2,1,1)}=X_{(3,2)}$ = [2,5]×[4,7] $X^{(2,2,1,1)}=X_{(3,1)}$ = [2,5]×[5,7]
$X^{(1,2,1,2)}=X_{(3,4)}$ = [2,5]×{7} $X^{(2,2,1,2)}=X_{(3,4)}$ = [2,5]×{7}
$X^{(1,2,2,1)}=X_{(1,2)}$ = [1,5]×[4,7] $X^{(2,2,2,1)}=X_{(0,1)}$ = [0,5]×[5,7]
$X^{(1,2,2,2)}=X_{(1,4)}$ = [1,5]×{7} $X^{(2,2,2,2)}=X_{(0,4)}$ = [0,5]×{7}

The solution set X is the union of these sets. All of them are Cartesian products of convex subsets of X_1 and X_2, respectively, and are themselves convex. We see that many sets $X^{\underline{k}}$ are contained in others but there are three sets which are maximal with respect to set inclusion: $S_1:=X^{(2,2,2,1)}=X_{(0,1)}$=[0,5]×[5,7], $S_2:=X^{(1,2,2,1)}=X_{(1,2)}$=[1,5]×[4,7], and $S_3:=X^{(1,1,1,1)}=X_{(2,0)}$=[3,5]×[0,7]. X can be represented as the union of these three sets only (see figure 1). It is clear that the union in characterization 2 may be restricted to include only those sets which are maximal among all $X^{\underline{k}}$. The following proposition states the converse of this statement.

Proposition: No set which is maximal among all $\underline{X}^{\underline{k}}$ can be removed from the union in characterization 2 without making the characterization invalid.

Proof: The sets $\underline{X}^{\underline{k}}$ are dual ideals in the sublattice U of $X_1 \times \cdots \times X_n$. We shall show that this implies that if some set $\underline{X}^{\underline{k}}$ is contained in the union of several other sets $S_1, S_2, \ldots, S_d$ (and can hence be removed from the union in characterization 2) then it must be contained in one of these sets, and hence it cannot be maximal. Assume the contrary: for each set S_e, $e=1,2,\ldots,d$, there is an element $\underline{x}_e \in \underline{X}^{\underline{k}} \setminus S_e$. The the infimum $\underline{x}_0$ of these elements $\underline{x}_e$ is also contained in $\underline{X}^{\underline{k}}$ but it is contained in none of the S_e, since $\underline{x}_e \geq \underline{x}_0 \in S_e$ would imply $\underline{x}_e \in S_e$.

Thus it makes sense to speak of the minimal set $\underline{K}' \subseteq \{1,\ldots,n\}^m$ for which characterization 2 is still true when the index $\underline{k}$ ranges only over $\underline{K}'$. Except for equal sets $\underline{X}^{\underline{k}}$, this set $\underline{K}'$ is unique.

We have shown:

Characterization 2':

$$X = \bigcup_{\substack{\underline{k} \in \{1,\ldots,n\}^m, \text{ and} \\ \underline{X}^{\underline{k}} \text{ is a maximal set}}} \underline{X}^{\underline{k}}.$$

The $\underline{X}^{\underline{k}}$ which are empty can of course all be omitted from the union. If $U=\emptyset$ then $X=\emptyset$ and all $\underline{X}^{\underline{k}}$ are empty as well; when this case is excluded, then those $\underline{k}$, for which $\underline{X}^{\underline{k}} = \emptyset$, can be easily identified: if $l_{ik_i} \supsetneq u_{k_i}$, then x_{k_i} cannot fulfill both defining conditions in the definition of the set $\underline{X}^{\underline{k}}$; on the other hand, if for each i $l_{ik_i} \subseteq u_{k_i}$, then $\underline{X}^{\underline{k}}$ is not empty. Therefore the set of those $\underline{k}$, for which $\underline{X}^{\underline{k}}$ contains an element is

$$\underline{K} = K_1 \times K_2 \times \ldots \times K_m,$$

where

$$K_i = \{k \mid l_{ik} \subset u_k\} = \{k \mid r_{ik}=1\} .$$

The sets K_i are of course just another way to describe the "1" elements in the rows of the 0-1-matrix R.

If each set $\underline{X}^k$ has a smallest element $\underline{l}^k$ (in the partial order on the lattice $X_1 \times \cdots \times X_n$), i. e. if the sets $u_j \setminus l_{ij}$ have smallest elements, then these smallest elements $\underline{l}^k$ correspond to their sets $\underline{X}^k$ one-to-one by the formula

$$\underline{X}^k = \{ \underline{x} \mid \underline{x} \epsilon U \text{ and } \underline{x} \geq \underline{l}^k \},$$

or

$$\underline{X}^k = \{ \underline{x} \mid \underline{l}^k \leq \underline{x} \leq \underline{u} \}, \text{ if the set U has a greatest element } \underline{u}.$$

The above question for a representation of X as the union of the minimal number of sets in the form of cartesian products is then equivalent to asking for <u>all minimal solutions</u> of (1), which are just the elements $\underline{l}^k$ for $\underline{k} \epsilon \underline{K}'$.

In the given example, the elements (0,5), (1,4), and (3,0) are the minmal solutions.

We have seen (formula (4)) that all sets $X^{\underline{k}}$ are of the form $X_{\underline{p}}$ for some sequence $\underline{p}$. Hence and from characterization 2 follows

Characterization 3:

$$X = \bigcup_{\substack{\underline{p}\in\{0,1,\ldots,m\}^n, \\ \text{and } X_{\underline{p}}\subseteq X}} X_{\underline{p}}.$$

Again, like above, we can prove that the maximal sets among those $X_{\underline{p}}$ with $X_{\underline{p}}\subseteq X$ are essential but all other $X_{\underline{p}}$ can be omitted, and we have

Characterization 3':

$$X = \bigcup_{\substack{\underline{p}\in\{0,1,\ldots,m\}^n, \\ X_{\underline{p}}\subseteq X, \text{ and} \\ X_{\underline{p}} \text{ is maximal}}} X_{\underline{p}}.$$

Here, "maximal" means maximal with respect to set inclusion among all sets $X_{\underline{p}}$ which are subsets of the solution set X. The set of subsets $X_{\underline{p}}$ over which the union in the above formula is taken is equal to the set of subsets $X^{\underline{k}}$, $\underline{k}\in\underline{K}'$, that are used in characterization 2'.

These characterization 3 and 3' of X are dual to the characterizations 2 and 2', respectively, in the sense that the sequences $\underline{k}$ which were used there were sequences assigning a column index k_i to every row index i, whereas the sequences $\underline{p}$ used here specify a row index p_j (or no row index, if p_j=0) for each column j.

The question left is how can it be decided whether for a given sequence $\underline{p}$ the set $X_{\underline{p}}$ is a subset of the solution set X. We shall answer this question in terms of the matrices $R(\underline{x})$.

From (4) it follows that $r_{p_j j}(\underline{x})=1$ for each j, if $\underline{x} \in X_{\underline{p}}$, but also $r_{ij}(\underline{x})=1$ for all i for which $l_{ij} \subseteq l_{p_j j}$, i. e. $R(\underline{x})$ is greater than or equal to the matrix $R_{\underline{p}}$ defined by

$$(R_{\underline{p}})_{ij}=1 \iff l_{ij} \subseteq l_{p_j j}.$$

If $R_{\underline{p}}$ is not contained in the matrix R, this means that the requirement $x_j \in l_{p_j j}$ is in conflict with the requirement $x_j \in u_j$, for some j, and $X_{\underline{p}}$ is empty. Otherwise there is always some $\underline{x} \in X_{\underline{p}}$ such that $R(\underline{x})=R_{\underline{p}}$. (The definition of $R_{\underline{p}}$ ensures that $(R_{\underline{p}})_{ij}=1$ if $l_{ij}=\emptyset$.) Thus, if the columns of $R_{\underline{p}}$ do not cover the set $\{1,\ldots,m\}$, then $X_{\underline{p}} \not\subseteq X$. On the other hand, if they do, then this holds a fortiori for all matrices $R(\underline{x})$, $\underline{x} \in X_{\underline{p}}$, and $X_{\underline{p}} \subseteq X$.

Like in the case of the sets $X^{\underline{k}}$, the sequences $\underline{p}$ for which the sets $X_{\underline{p}}$ are empty and can surely be excluded, can be described concisely: We have just seen that $X_{\underline{p}}$ is non-empty if and only if $R_{\underline{p}} \leq R$, i. e. if and only if $l_{p_j j} \subset u_j$ for all j, i. e. if $r_{p_j j}=1$ for all j. Therefore the set of those $\underline{p}$ for which $X_{\underline{p}}$ contains an element is

$$P_1 \times P_2 \times \cdots \times P_n,$$

where

$$P_j = \{\, p \mid l_{pj} \subseteq u_j \,\} = \{\, p \mid r_{pj}=1 \,\}.$$

The sets P_j are again just another way to describe the matrix R.

Note that $\underline{p} \in P_1 \times \cdots \times P_n$ does not imply that $X_{\underline{p}} \subseteq X$.

Thus if the sets $M_i^{(j)} := \{ 1 \leq i' \leq m \mid l_{i'j} \subseteq l_{ij} \}$, $0 \leq i \leq m$, are the possible column sets corresponding to column j, $1 \leq j \leq n$, then $\underline{P}$, the set over which the union in characterization 3 is taken, can be characterized by a kind of generalized set covering problem in the following way:

<u>Generalized set covering problem</u>

For each $1 \leq j \leq n$ a chain $\underline{M}^{(j)} = \{ M_0^{(j)}, M_1^{(j)}, \ldots, M_{m_j}^{(j)} \}$ of subsets of $\{1,\ldots,m\}$ is given.

The sequence $\underline{M}_{\underline{p}} = (M_{p_1}^{(1)}, M_{p_2}^{(2)}, \ldots, M_{p_n}^{(n)})$, which is associated with the sequence $\underline{p} = (p_1, p_2, \ldots, p_n)$, where $0 \leq p_j \leq m_j$, is called a <u>covering</u> if

$$\bigcup_{1 \leq j \leq n} M_{p_j}^{(j)} = \{1,\ldots,m\}.$$

(Here, a set $\underline{M}$ of sets is called a <u>chain</u> if in each pair of sets from $\underline{M}$ one set contains the other.)

The feasible solutions of the ordinary set covering problem correspond to the case: $m_j = 1$ and (w. l. o. g.) $M_0^{(j)} = \emptyset$, for all j.

Frequently, we shall also call the sequences $\underline{p}$ coverings.

With this definition, $\underline{P}$ is just the set of coverings, and we can reformulate characterization 3 as follows:

<u>Characterization 4:</u>

$$X = \bigcup_{\underline{p} \text{ is a covering}} X_{\underline{p}}.$$

Since we want to express also characterization 3' combinatorially, we are interested in coverings $\underline{p}$, for which $X_{\underline{p}}$ is maximal. The inclusion order for the sets $X_{\underline{p}}$ can be expressed in terms of the sequences $\underline{p}$ as follows:

$$
\begin{aligned}
X_{\underline{p}} \subseteq X_{\underline{q}} \quad &\Leftrightarrow \quad \text{for all } j\text{: } u_j \setminus l_{p_j j} \subseteq u_j \setminus l_{q_j j} \quad \Leftrightarrow \\
&\Leftrightarrow \quad X_{\underline{p}} = \emptyset \text{ or for all } j\text{: } l_{p_j j} \supseteq l_{q_j j} \quad \Leftrightarrow \\
&\Leftrightarrow \quad X_{\underline{p}} = \emptyset \text{ or for all } j\text{: } M_{q_j}^{(j)} \subseteq M_{p_j}^{(j)} \, .
\end{aligned}
$$

Thus, we are interested in coverings which are minimal with respect to the following partial order:

$$
\left(M_{p_1}^{(1)}, \ldots, M_{p_n}^{(n)}\right) \leq \left(M_{q_1}^{(1)}, \ldots, M_{q_n}^{(n)}\right)
$$
$$
\text{iff} \quad \text{for all } j\text{: } M_{p_j}^{(j)} \subseteq M_{q_j}^{(j)} \, .
$$

We shall also speak of the corresponding sequences $\underline{p}$ and $\underline{q}$ as ordered by the relation $\leq$, although this ordering is in general only a preorder since different sequences $\underline{p}$ and $\underline{q}$ may yield the same sequences $\underline{M}_{\underline{p}}$ and $\underline{M}_{\underline{q}}$. Thus, if the sets $M_i^{(j)}$ with $l_{ij} \supseteq u_j$, which yield empty sets $X_{\underline{p}}$, are removed from the chains $\underline{M}^{(j)}$ in the given data of the generalized set covering problem, we get

<u>Characterization 4'</u>:

$$
X = \bigcup_{\underline{p} \text{ is a minimal covering}} X_{\underline{p}} \, .
$$

<u>Example</u>:

For the system of equations of example 1, we obtain the following sets $M_p^{(j)}$:

	$M_0^{(1)}$	$M_1^{(1)}$	$M_2^{(1)}$	$M_3^{(1)}$	$M_4^{(1)}$	$M_0^{(2)}$	$M_1^{(2)}$	$M_2^{(2)}$	$M_3^{(2)}$	$M_4^{(2)}$
1		X	X	X			X			X
2			X				X	X		X
3			X	X			X	X	X	X
4	X	X	X	X	X					X

An X in line i means that the element i is contained in the corresponding set. Since the element 4 is alway covered by some $M_p^{(1)}$, it need not be considered. After eliminating it from the table and after renumbering the sets $M_p^{(j)}$, $0 \leq p \leq 4$, in increasing order and identifying equal sets (for each fixed j) the following table is obtained:

	$M_0^{(1)}$	$M_1^{(1)}$	$M_2^{(1)}$	$M_3^{(1)}$	$M_0^{(2)}$	$M_1^{(2)}$	$M_2^{(2)}$	$M_3^{(2)}$
1		X	X	X				X
2				X			X	X
3			X	X		X	X	X

which can be represented in a compact form as follows:

i | j:	1	2
1	1	3
2	3	2
3	2	1

For example, the entry "2" in the third line, first column, means that $M_2^{(1)}$ contains all those elements i for which the i^{th} entry in the first column is not greater than two, i. e. the elements 1 and 3.

The minimal coverings of this generalized set covering problem are (3,0), (1,2), and (0,3), which yields by characterization 4' the following minimal representation of the solution set:

$$X = X_{(3,0)} \cup X_{(1,2)} \cup X_{(0,3)} =$$
$$= [3,5]\times[0,7] \cup [1,5]\times[4,7] \cup [0,5]\times[5,7] .$$

Thus, the minimal elements of the solution set are (3,0), (1,4), and (0,5) (see figure 1). (Remember that the sets have been renumbered, and therefore $X_{(3,0)}$ is the set which was formerly called $X_{(2,0)}$, and so on; see the values given after equation (4) for this example.)

There is a variation of characterization 4, which is useful for enumerating all solutions of the system (1), since it represents X as a disjoint union of rectangular sets (sets in the form of Cartesian products).

Assume that for all j the sets l_{ij}, $0 \leq i \leq m$, are relabeled such that $l_{0j} \subset l_{1j} \subset \cdots \subset l_{m_j j} \subset u_j$ (equal sets l_{ij} occur only once in this sequence), and that $l_{ij} \supseteq u_j$ (i. e. $i \notin P_j$) if $i > m_j$. Then we can write the set $u_j \setminus l_{pj}$, which occurs in formula (4), as a disjoint union

$$(5) \qquad u_j \setminus l_{pj} = (l_{p+1,j} \setminus l_{pj}) \cup (l_{p+2,j} \setminus l_{p+1,j}) \cup \ldots \cup (u_j \setminus l_{m_j j}),$$

provided that the set $(u_j \setminus l_{pj})$ is not empty, i. e. $p \leq m_j$.

We define for each sequence $\underline{p}=(p_1,\ldots,p_n)$ with $0 \leq p_j \leq m_j$

$$X_{(\underline{p})} := (l_{p_1+1,1} \setminus l_{p_1 1}) \times (l_{p_2+1,2} \setminus l_{p_2 2}) \times \cdots (l_{p_n+1,n} \setminus l_{p_n n}),$$

where $l_{m_j+1,j}$ is set to u_j, for all j.

Then it follows immediately from (4) and (5), that $X_{\underline{p}}$ is representable as a disjoint union:

$$X_{\underline{p}} = \bigcup_{\underline{p} \leq \underline{q} \leq \underline{m}} X_{(\underline{q})}, \text{ where } \underline{m}=(m_1,\ldots,m_n).$$

Characterization 5:

X, the set of solutions of (1), is representable as the disjoint union of non-empty sets:

$$X = \bigcup_{\substack{\underline{p} \in P \\ \underline{p} \leq \underline{m}}} X_{(\underline{p})} \; .$$

Proof: $\underline{p} \in \underline{P}$ and $\underline{p} \leq \underline{q}$ implies $\underline{q} \in \underline{P}$ ($\underline{P}$ is the set of coverings). That X is thus representable follows then from characterization 4 and the preceding equation. That the sets $X_{(\underline{p})}$ are non-empty and that the union is disjoint follows from the definition of $X_{(\underline{p})}$.

Note the sets in this representation are not the minimal number of disjoint rectangular sets (sets in the form of Cartesian products of intervals).

Example

From the first table of the sets $M_p^{(j)}$ given for the data of example 1 after characterization 4' we obtain after identifying equal sets l_{ij} and reordering and renumbering the sets l_{ij} in increasing order in each column:

	$M_0^{(1)}$	$M_1^{(1)}$	$M_2^{(1)}$	$M_3^{(1)}$	$M_0^{(2)}$	$M_1^{(2)}$	$M_2^{(2)}$	$M_3^{(2)}$	$M_4^{(2)}$
1		X	X	X				X	X
2				X			X	X	X
3			X	X		X	X	X	X
4	X	X	X	X					X
l_{ij}	$\emptyset$	[0,1)	[0,2)	[0,3)	$\emptyset$	[0,1)	[0,4)	[0,5)	[0,7)

$m_1=3$, $m_2=4$, $l_{41}=u_1=[0,5]$, $l_{52}=u_2=[0,7]$.

The set of all coverings is

$\underline{P}$ = { (3,0), (3,1), (3,2), (3,3), (3,4), (2,2), (2,3), (2,4), (1,2), (1,3), (1,4), (0,3), (0,4) },

and we get X represented as

$$X = [3,5]\times[0,1) \cup [3,5]\times[1,4) \cup [3,5]\times[4,5) \cup [3,5]\times[5,7) \cup [3,5]\times\{7\} \cup [2,3)\times[4,5) \cup [2,3)\times[5,7) \cup [2,3)\times\{7\} \cup [1,2)\times[4,5) \cup [1,2)\times[5,7) \cup [1,2)\times\{7\} \cup [0,1)\times[5,7) \cup [0,1)\times\{7\}.$$

3. Algorithms for enumerating the minimal solutions of the generalized set covering problem

In this section, two algorithms for enumerating the minimal solutions of the generalized set covering problem are presented, and one of them will be specialized for the set covering problem in section 4. The execution of the algorithms is illustrated by some examples. The set covering problem, i. e. the problem of minimizing a certain function over all possible coverings, is a well-known problem. Since it is NP-hard [Garey and Johnson 1979], the methods for its exact solution are basically (implicit) enumeration methods, and therefore ideas from those methods can be used in the algorithms of this section. For example, in certain cases reductions can be applied to the data of a given problem to transform it into a smaller equivalent problem. The reduction rules shall not be treated here; they are discussed in almost every text on covering problems (e. g. Garfinkel and Nemhauser [1972, section 8.3: reductions, in chapter 8: the set covering and partitioning problems]; or Sysło, Deo, and Kowalik [1983, section 2.2: covering algorithms]).

3.1. An algorithm for enumerating the minimal solutions of the generalized set covering problem

In the following we shall assume that for all j the sets $M_p^{(j)}$ for which $l_{pj} \geq u_j$ are removed from $\underline{M}^{(j)}$, that the remaining sets in $\underline{M}^{(j)}$ are renumbered such that $M_0^{(j)} \subset M_1^{(j)} \subset \ldots \subset M_{m_j}^{(j)}$, and that $M_0^{(j)} = \emptyset$. (Otherwise the elements in $M_0^{(j)}$ are always covered and can be eliminated from the ground set $\{1,\ldots,m\}$).

Each set $M_p^{(j)}$ for p>0 has a <u>critical set</u> of elements which distinguish this set from the next smaller set in $\underline{M}^{(j)}$: $M_p^{(j)} \setminus M_{p-1}^{(j)}$. If in a covering $\underline{p}=(p_1,\ldots,p_n)$ each element in some critical set $M_{p_j}^{(j)} \setminus M_{p_j-1}^{(j)}$ is covered by at least one other set than $M_{p_j}^{(j)}$, then p_j can be reduced and $\underline{p}$ cannot be minimal. It is also clear that the converse of this condition is sufficient for minimality.

Thus, we obtain

<u>Lemma 1</u>: (<u>criterion for minimality</u> of a covering <u>p</u>):

A covering $\underline{p}=(p_1,\dots,p_n)$ is a minimal covering if and only if for each j with $p_j>0$ the critical set $M_{p_j}^{(j)}\setminus M_{p_j-1}^{(j)}$ contains an element which is not contained in any other $M_{p_{j'}}^{(j')}$ $(1\leq j'\leq n, j'\neq j)$.

The algorithm is based on characterization 4', and it is a variation of the following program, which is a set of nested loops, which enumerates the sequences $\underline{p}=(p_1,\dots,p_n)$ in lexicographic order and selects the minimal coverings.

```
for p1 from 0 to m1 do
    for p2 from 0 to m2 do
        .
          .
            .
              for pn from 0 to mn do
                  if (p1,...,pn) is a covering
                                      and if it is minimal
                      then print (p1,...,pn);
                  end if
              end for;
            .
          .
        .
    end for;
end for.
```

The following recursive procedure represents one level in the set of nested <u>for</u>-loops of the above program. The test for minimality is executed as early as possible, and the algorithm attempts to exclude the non-coverings at an early stage.

```
procedure GSCP1(j);
  { At this level, (p1,...,pj-1) are fixed and the procedure will select
    the values for pj. }

    for k from j to n do
        pmax_k := m_k;
        { pmax_k will be an upper bound on the possible values of p_k. }
    end for;
    for all j' with 1≤j'<j and p_j'>0 do
        cs := M_pj'^(j') \ M_pj'-1^(j');
              { This is the critical set corresponding to j'. }
        cs := cs \      U      M_pj"^(j");
                    1≤j"<j
                     j"≠j'
              { remove the elements which are already covered }
        for k from j to n do
            if M_mk^(k) ⊇ cs then
                q_kj' := the smallest p, such that M_p^(k) contains cs;
                { M_pk^(k) must not cover cs, therefore p_k<q_kj': }
                pmax_k := min(pmax_k, q_kj'-1);
            end if;
        end for;
    end for;

    uncovered := {1,2,...,m} \      U      M_pj'^(j');
                                1≤j'<j
            { This is the set of the elements uncovered so far. }
    if     U     M_pmax_k^(k)  ⊇  uncovered   then
       j≤k≤n
        if j<n then
            pj := 0;    GSCP1(j+1);
            for pj from 1 to pmax_j do
                cs := M_pj^(j) \ M_pj-1^(j);
                    { This is the critical set corresponding to j. }
                if cs∩uncovered ≠ ∅ then
                    GSCP1(j+1);
                end if;
            end for;
        else { j=n }
            pn := the smallest p such that M_p^(n) ⊇ uncovered;
            print (p1,...,pn);
        end if;
    else { There can be no minimal covering starting with (p1,...,pj-1). }
    end if;
end GSCP1;
```

Example 2:

Consider the following generalized set covering problem, which is presented in tabular form as was already explained in connection with example 1:

i: | j:	1	2	3
1:	1	4	5
2:	1	2	6
3:	2	3	4
4:	3	3	1
5:	3	1	2
6:	-	4	3

(The entry "-" in the first column means that the element 6 is not covered by any set $M_p^{(1)}$.)

Figure 2 shows the solution of this problem with the algorithm GSCP1.

There are six minimal coverings $\underline{p}$; In lexicographic order, as they are produced by the procedure, they are:

(0,0,6)
(0,2,5)
(0,4,0)
(1,0,4)
(1,3,3)
(2,0,3)

Example 3:

The generalized set covering problem

i: | j:	1	2	3	4
1:	1	-	2	1
2:	1	1	3	-
3:	2	1	1	-

has six minimal coverings $\underline{p}$; In lexicographic order, as they are produced by the procedure GSCP1 (see figure 3), they are:

(0,0,3,0)
(0,1,0,1)
(0,1,2,0)
(1,0,1,0)
(1,1,0,0)
(2,0,0,0)

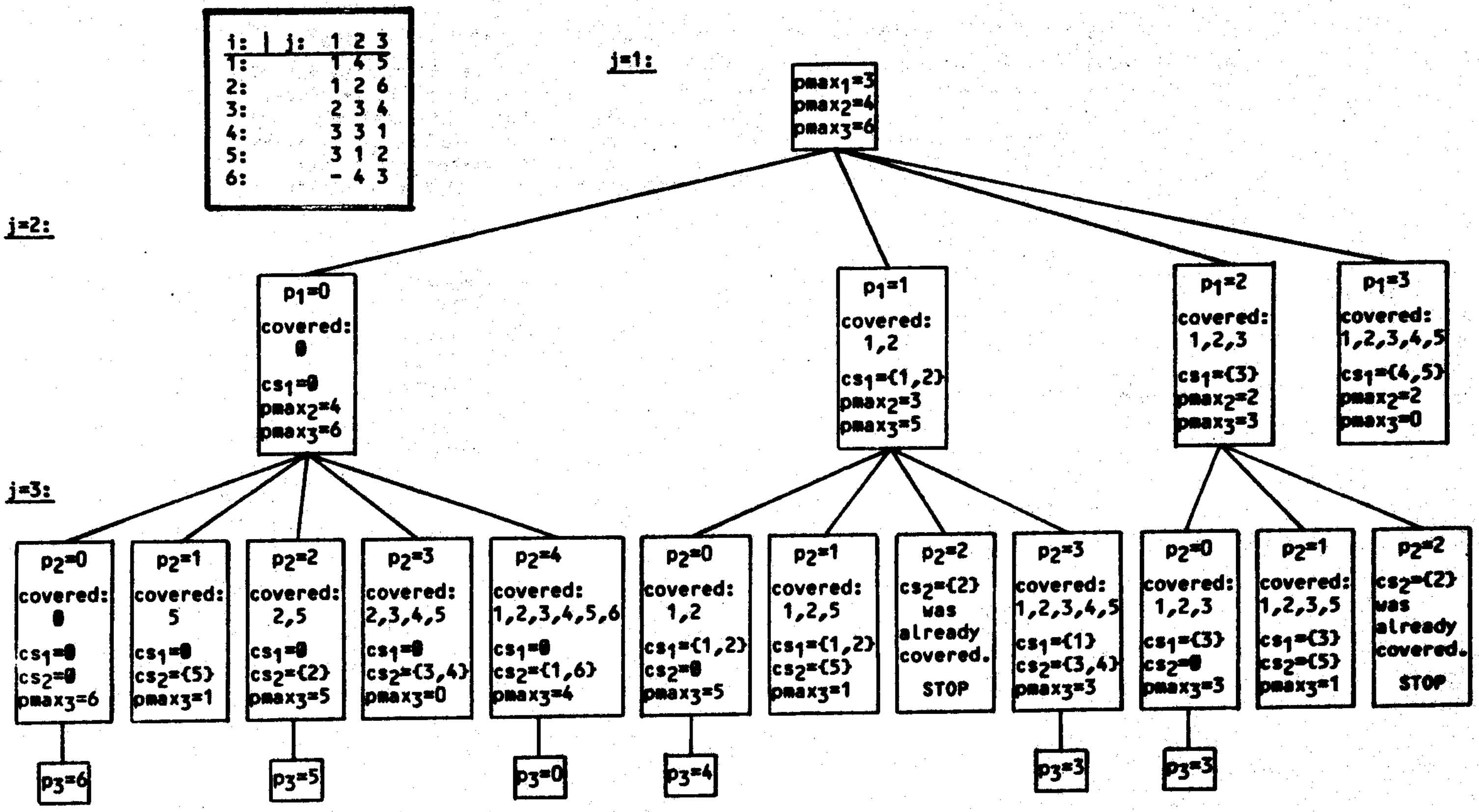

Figure 2 - the tree of recursive calls of the procedure GSCP1 for example 2

If the procedure has determined that there can be no complete covering with the sets restricted by $pmax_j$, $pmax_{j+1}$, ..., $pmax_n$, then this is not otherwise indicated except that the respective node in the tree has no sons. If the procedure excludes some value of p_j because the corresponding critical set is already covered, then this is written in the nodes. (This occurs twice in this example. Actually, as the procedure is written, this test is executed one level higher than it is shown in the diagram.)

i:	j: 1	2	3	4
1:	1	-	2	1
2:	1	1	3	-
3:	2	1	1	-

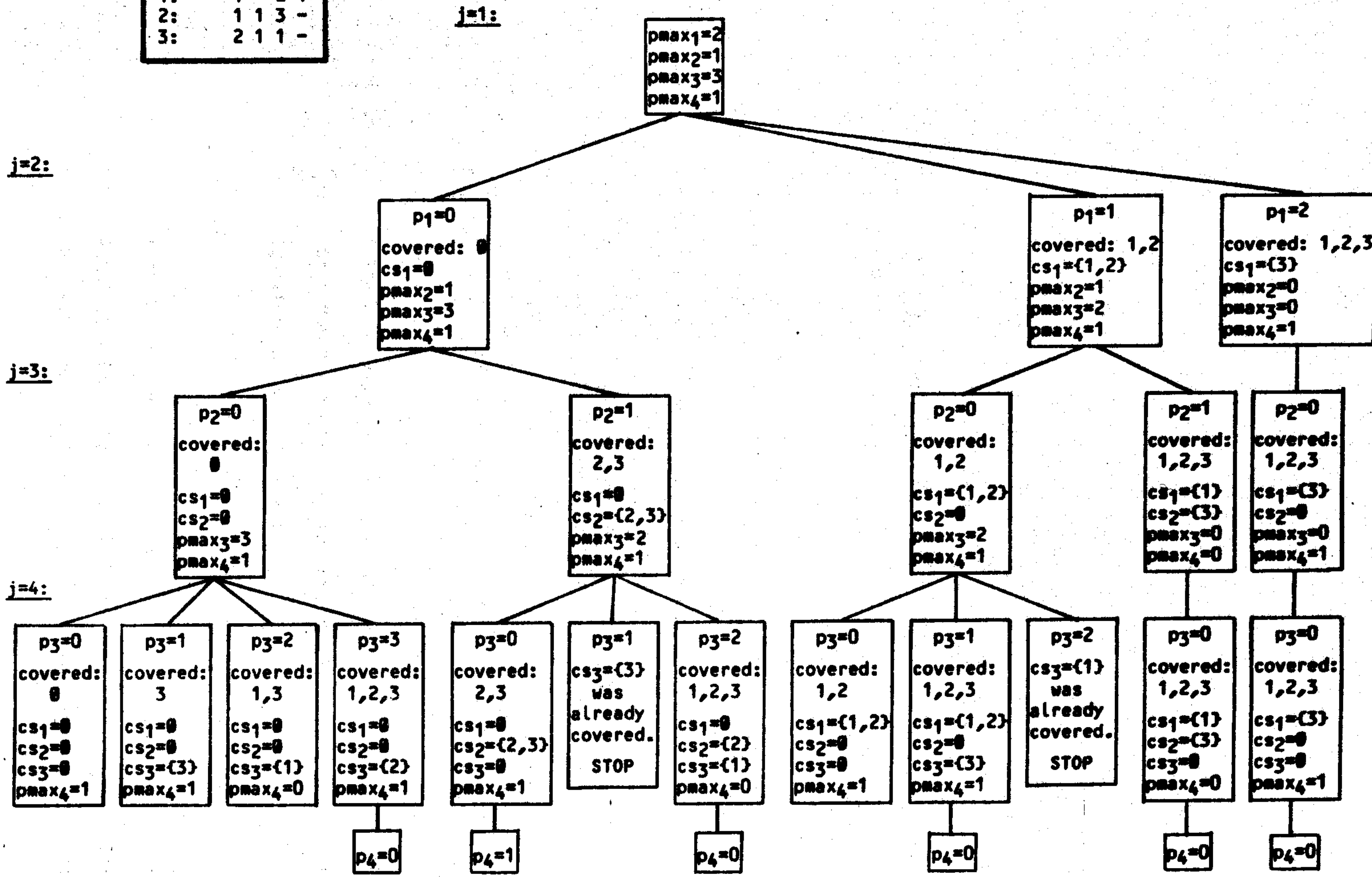

Figure 3 - the tree of recursive calls of the procedure GSCP1 for example 3
The same conventions as in figure 2 are used.

The procedure has been written in a style which rather aims to be understandable than efficient. If it should be implemented on a computer, many simplifications can be made, and the data structures for the representation of the sets should be chosen with care. For example, the critical sets cs need not be recomputed each time, but are readily representable as an array cs_{pj} of lists with little storage expense. An array $count_i$ could be maintained which counts by how many sets each element i has been covered so far. To simplify the determination of $q_{kj'}$, there could be a two-dimensional array $pmin_{ij}$ which stores for each i and j the smallest p such that $i \in M_p^{(j)}$ (or infinity, if there is no such p).

With these modifications, the loop " for all j' with $1 \leq j' < j$ and $p_{j'} > 0$ " could be implemented as follows:

```
for all j' with 1≤j'<j and pj'>0 do
    for k from j to n do
        q := 0;
        for all i in cs_pj',j' do
            if count_i=1 then
                q := max(q, pmin_ij');
            end if;
        end for;
        pmax_k := min(pmax_k, q-1);
    end for;
end for;
```

This part of the program will then take at most $O(mn^2)$ steps, taking the upper bound of m for the number of elements of the critical sets. Nevertheless, the algorithm is not efficient. If the test

$$\text{" } \underline{\text{if}} \bigcup_{j \leq k \leq n} M_{pmax_k}^{(k)} \supseteq \text{uncovered "}$$

is passed it is not guaranteed that a minimal covering starting with $(p_1, p_2, \ldots, p_{j-1})$ will be found, except in the case where all critical sets are of cardinality 1. In this last case the algorithm cannot run into dead ends and the time can be bounded as follows: When the algorithm has been successful in finding a minimal covering p, then at each level j (from 1 to n-1) on the chain of recursive calls of the procedure, there have been at most $pmax_j+1 \leq m+1$ recursive calls of GSCP1(j+1). One of them has lead to the minimal covering p, the rest of the calls have either returned after the failure of the test

$$\text{" } \underline{\text{if}} \bigcup_{j \leq k \leq n} M_{pmax_k}^{(k)} \supseteq \text{uncovered "}$$

without any further recursive subcalls or they have lead to other minimal coverings. Therefore, at most (n-1)(m+1) plus one (for the call at level j=1) calls can be booked on the account of the minimal covering p, each call taking at most $O(n^2)$ elementary steps. This yields a time bound of $O(mn^3)$ per detected minimal covering. This time bound, which holds even only in this special case, is very high, especially when there are many minimal coverings. In this latter case, the anticipated test for covering, which takes the most part of the work in the procedure, might be omitted, since it will be satisfied anyway most of the times. In detail this means that the two loops " for k from j to n ", where the $pmax_k$ are determined, are executed only " for k:=j ", and the test

$$\text{" } \underline{\text{if}} \bigcup_{j \leq k \leq n} M_{pmax_k}^{(k)} \supseteq \text{uncovered "}$$

is omitted except when j=n. The procedure takes now only O(mn) elementary steps. It is also conceivable to use both versions of the procedure together: the abbreviated one for small j, and the extended one for the last few levels (j=n-s, n-s+1, ..., n-1), when the work for the extended version is not so large (O(mns)), but the time savings through the anticipated test can still be considerable.

The sequence in which the p_j are considered in $(p_1,...,p_n)$ may be chosen arbitrarily, and it certainly plays a role in the performance of the algorithm.

3.2. Another algorithm for enumerating the minimal solutions of the generalized set covering problem

There is an algorithm based on characterization 2', which is dual to the algorithm in the preceding section. In principle one approaches the problem as in the previous algorithm (which is based on characterization 4'), generating all sequences $\underline{k}\in\underline{K}$ in a systematic way, and testing at the end whether the generated solution set is maximal or not (i. e. whether the generated covering is minimal or not). However, since each minimal covering is to be enumerated only once, there should be a one-to-one correspondence between the sequences $\underline{k}$ generated and the minimal coverings.

Let a minimal covering $\underline{p}=(p_1,\ldots,p_n)$ be given. With this covering we can uniquely associate a set of pairs

$$\underline{k}(\underline{p}) := \{ (i_j,j) \mid p_j>0 \},$$

where, for each j with $p_j>0$:

(8) i_j is the smallest number (the critical element) among the elements of the critical set $M_{p_j}^{(j)} \setminus M_{p_j-1}^{(j)}$ which are covered only once.

(That there is such an index i_j follows from the minimality of the covering.)

(Since the ordering of the elements i is irrelevant, this condition is somewhat arbitrary.)

Example:

In example 2, (1,3,3) and (1,0,4) are minimal coverings, which can be represented as follows:

```
     1 3 3              1 0 4
 1:  X              1:  X
 2:  X o            2:  X
 3:    X            3:      X
 4:    X o          4:      o
 5:    o o          5:      o
 6:      X          6:      o
```

$\underline{k}(\underline{p})=\{(1,1),(3,2),(6,3)\}$ $\underline{k}(\underline{p})=\{(1,1),(3,3)\}$

$\underline{k}_{\underline{p}}=(1,0,2,0,0,3)$ $\underline{k}_{\underline{p}}=(1,0,3,0,0,0)$

Here the three columns represent the elements of the three sets $M_{p_1}{}^{(1)}$, $M_{p_2}{}^{(2)}$, and $M_{p_3}{}^{(3)}$, and the X's are the critical elements of those sets. In the first example, each set contains only one critical element that is covered only once, and hence $\underline{k}(\underline{p})$ is uniquely determined. In the second example, the set $M_{p_1}{}^{(1)}$ contains two such critical elements, and hence (1,1) could be replaced by (2,1) in the set $\underline{k}(\underline{p})$ if condition (8) did not ensure uniqueness by requiring to select the smallest possible i_j.

For each i, $1 \leq i \leq m$, there is at most one pair (i,j) in $\underline{k}(\underline{p})$. Thus, $\underline{k}(\underline{p})$ could be regarded as a partial sequence $((i,k_i))$ of elements from $\{1,\ldots,n\}$. These partial sequences are also given in the above example, where entries k_i which are not present are represented as zeroes. With such a partial sequence $\underline{k}$, a unique set $X^{\underline{k}}$ can be associated just as with ordinary sequences $\underline{k} \in \{1,\ldots,n\}^m$ (compare (3)):

$$(3') \quad X^{\underline{k}} := \{ (x_1,\ldots,x_n) \mid \text{for all } 1 \leq j \leq n: x_j \in u_j; \text{ and for all } (i,k_i) \in \underline{k}: x_{k_i} \notin l_{ik_i} \}.$$

Conversely, by formula (4), a sequence $\underline{p} \in \{0,1,\ldots,m\}^n$ is <u>generated</u> by a partial sequence $\underline{k}$ and the set $X^{\underline{k}}$ just as by a full sequence $\underline{k}$. Thus, a one-to-one correspondence between those sequences $\underline{k}$ and the desired solution sets has been established. However, not for all partial sequences $\underline{k}$, the sets $X^{\underline{k}}$ are solution sets, or, equivalently, not every sequence $\underline{p}$ generated by a partial sequence $\underline{k}$ is a covering. Even if $\underline{p}$ is a covering, it need not be minimal, or $\underline{k}$ need not conform to condition (8). Only when all these conditions are satisfied we can accept $\underline{k}$. Every partial sequence $\underline{k}$ satisfying these conditions can be extended to a full sequence by choosing each of the remaining k_i to be any j for which $M_{p_j}{}^{(j)}$ covers i. The set $X^{\underline{k}}$ does not change when this extension is made. The set of all those extended sequences $\underline{k}$ is just the minimal set $\underline{K}'$ (cf. characterization 2' and the discussion preceding it).

In the following algorithm, which is presented in a way analogous to GSCP1, the partial sequences $\underline{k}$ are represented as sequences $(k_1,\ldots,k_m) \in \{0,1,\ldots,n\}^m$, where k_i is set to 0 if k_i is not present in $\underline{k}$.

```
procedure GSCP2(i);
  { At this level, (k1,...,ki-1) are fixed and the procedure will select
    the values for ki.
    If i=m+1, however, the procedure tests whether the sequence p
    generated by k=(k1,...,km) is a minimal covering and whether k
    conforms to condition (8). }
    if i≦m then
        if iε   U   Mpj(j) then
              1≦j≦n
            ki := 0;    { i is already covered. }
            GSCP2(i+1);
        else
            for all ki from 1 to n with ki∉{k1,...,ki-1} do
                if iεMmki(ki) then
                    pki := the smallest p such that iεMp(ki);
                    if {i'<i|ki'≠0} ∩ Mpki(ki) = ∅ then
                        GSCP2(i+1);
                    else  { Mpki(ki) would cover the critical element of
                              some other critical set. }
                    end if;
                    pki := 0; { reset pki to its previous value }
                end if;
            end for;

            ki := 0;    yet-to-be-covered := yet-to-be-covered ∪ {i};
            GSCP2(i+1);
            yet-to-be-covered := yet-to-be-covered \ {i};
                                          { restore previous value }
        end if;
    else { i=m+1 }
        for all i'εyet-to-be-covered do
            if  i' ∉   U   Mpj(j)      { in this case p is not a covering; }
                     1≦j≦n
            or if   i' is contained in some critical set Mpj(j)\Mpj-1(j),
                    for some 1≦j≦n with pj>0, but there is no other j',
                    1≦j'≦n, j'≠j, such that i'εMpj'(j')
                      { in this case k does not conform to condition (8),
                        since i' is covered only once and hence the
                        critical element ij of Mpj(j) is i' or smaller,
                        and p has already been output before with kij=j. }
            then return from procedure;
            end if;
        end for;
        { (p1,...,pn) has passed all tests. }
        print (p1,...,pn);
    end if;
end GSCP2;
```

This procedure uses some global variables which must be initialized as follows:

```
program initialize-for-GSCP2;
     for j from 1 to n do
         p_j := 0;
     end for;
     yet-to-be-covered := Ø;   { The set yet-to-be-covered will contain
                                  those elements i which are left uncovered
                                  in the procedure at level i; these
                                  elements must eventually be tested for
                                  being covered.  }
     GSCP2(1);
end.
```

The tree of calls of the procedure for the data of example 3 is given in figure 4.

What has been said in connection with the programmin style and the implementation of GSCP1 on a computer applies also to GSCP2. The time can be bounded by O(mn) steps per call of the procedure and by O(m) for the final test (when i=m+1). The total time of the program can again only be bounded exponentially, even in relation to the produced output.

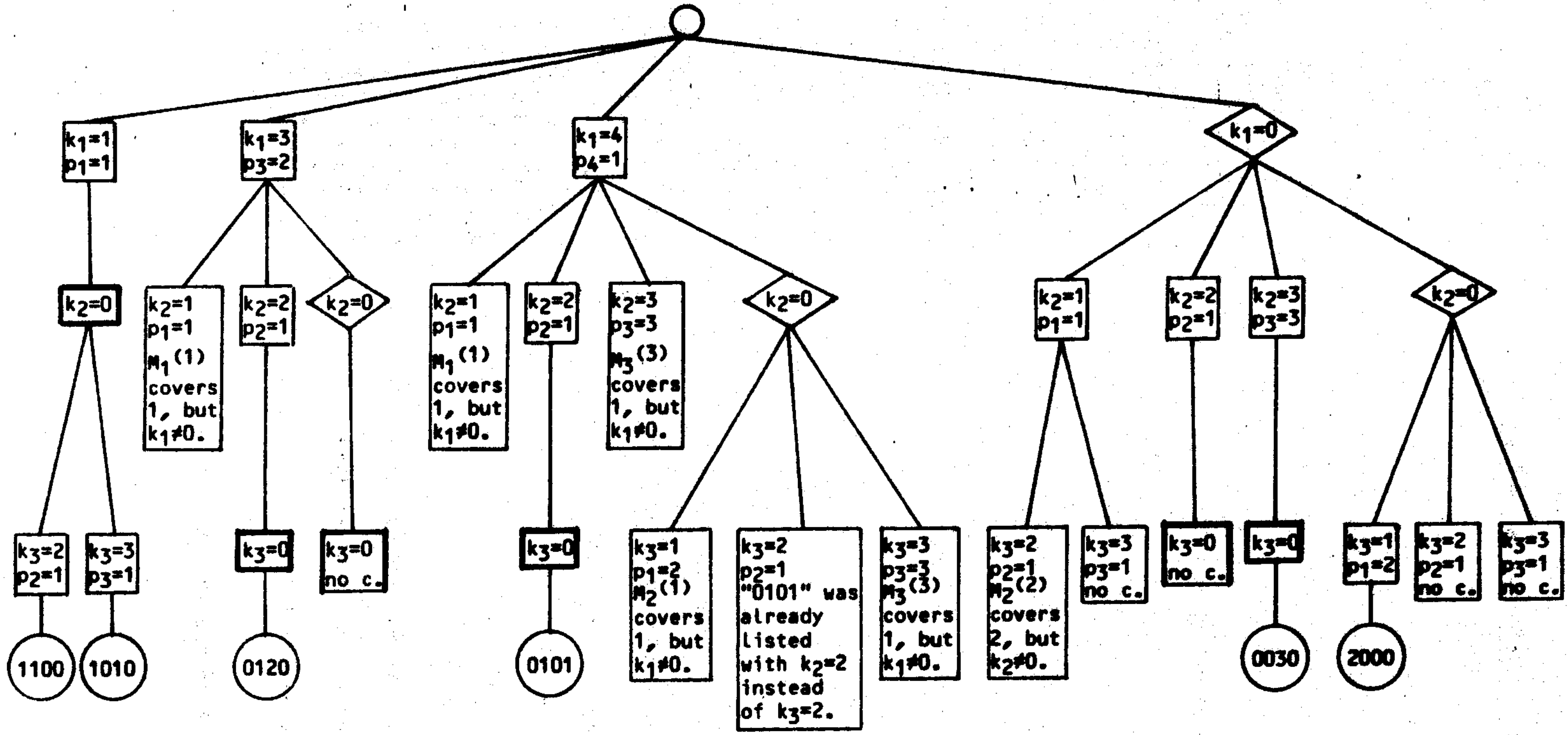

Figure 4 - the tree of recursive calls of the procedure GSCP2 for example 3

If the procedure sets $k_i := 0$, because the element i is already covered then the corresponding node is drawn in a thick rectangle. The parallelograms are the nodes where k_i is set to 0, but i is not covered yet, but is included into the set yet-to-be-covered. At the last level (i=3), these nodes are omitted (deviating from the procedure as given in the text), because if the element 3 is not covered at this level then it will not be covered at all.
At the last level, the test on the elements in the set yet-to-be-covered are executed. If the generated sequence p̲ is not a covering, then this is indicated as "no c.". In one case (when $\underline{k}=(4,0,2)$), the previously uncovered element 2 is the smallest critical element in the set $M_1^{(2)}$ but $k_2=0$, contradicting condition (8). The detected minimal coverings p̲ are given in the circles.

4. Special case (a) Maxsum (minsum, resp.) equations

Here we have $X_1=\ldots=X_n=B_1=\ldots=B_m=:B$, where $(B,\circledast,\leqq)$ is a linearly ordered group, and in (1) $f_{ij}(x) = a_{ij}\circledast x$ for given elements $a_{ij}\in B$. The most common examples are the real groups $(\mathbb{R},+,\leqq)$ and $(\mathbb{R}^+,\cdot,\leqq)$ and subgroups of them.

If we specialize the results of the preceding sections for this problem we obtain

$$I_{ij} = \{ x\in B \mid a_{ij}\circledast x=b_i \} = \{ a_{ij}^{-1}\circledast b_i \},$$

and, with $c_{ij}:=a_{ij}^{-1}\circledast b_i$,

$$u_{ij} = \{ x\in B \mid x\leqq c_{ij} \}, \text{ and}$$

$$l_{ij} = \{ x\in B \mid x<c_{ij} \}.$$

$$u_j = \bigcap_{1\leqq i\leqq m} u_{ij} = \{ x\in B \mid x\leqq c_j \}, \text{ where } c_j := \min_{1\leqq i\leqq m} c_{ij}.$$

For the sequences $\underline{p}$ with non-empty $X_{\underline{p}}$ (see the discussion after the characterizations 3 and 3') we obtain

$$P_j = \{ p \mid l_{pj}\subset u_j \} = \{ p \mid c_{pj}\leqq c_j \}.$$

Since always $c_{pj}\geqq c_j$, we can write

$$P_j = \{ p \mid c_{pj}=c_j \}.$$

In the generalized set covering problem referred to in characterization 4 there is only one set in each $\underline{M}^{(j)}=\{\emptyset,P_j\}$, apart from the empty set $M_0^{(j)}$.

$$M_j := \{ p \mid 1\leqq p\leqq m,\ c_{pj} = \min_{1\leqq i\leqq m} c_{ij} \},$$

writing M_j instead of $M_1^{(j)}$.

Therefore, in this case, the generalized set covering problem referred to in characterization 4 reduces to the

<u>Set covering problem</u>

For each $1 \leq j \leq n$ a subset M_j of $\{1,\ldots,m\}$ is given.

The set $P \subseteq \{1,2,\ldots,n\}$ is called a <u>covering</u> if

$$\bigcup_{p \in P} M_p = \{1,\ldots,m\}.$$

We are interested in coverings which are minimal with respect to set inclusion.

In contrast with the formulation of the generalized set covering problem, we have written sets $P \subseteq \{1,\ldots,n\}$ instead of sequences $\underline{p} \in \{0,1\}^n$.

Either of the algorithms for the generalized set covering problem described above can be specialized for the set covering problem, but there is a simplification only for the program GSCP2, which will be described now. In its formulation the sets K_i from the discussion after characterization 2' are used. We have (see also the remark after characterization 3'):

$$K_i = \{ j \mid c_{ij} \leq c_j \} = \{ j \mid c_{ij} = c_j \} = \{ j \mid i \in M_j \}.$$

A necessary and sufficient condition for the existence of a solution of the system (1) is that no set K_i is empty.

As a rather straightforward specialization of GSCP2 with slight variations in formulations, the following procedure should not be difficult to understand.

```
procedure SCP(i);
  { At this level, the values of k1,...,ki-1 are fixed and the procedure
    will select the values for ki, and Pi-1 is the set generated by
    (k1,...,ki-1,0,...,0).
    If i=m+1, however, the procedure tests whether the set Pm generated by
    k=(k1,...,km) is a minimal covering and k conforms to condition (8). }
    if i≤m then
        if Pi-1 ∩ Ki ≠ ∅ then
            ki := 0;    { i is already covered. }
            Pi := Pi-1;   SCP(i+1);
        else
            for all ki∈Ki do
                if {i'<i|ki'≠0} ∩ Mki = ∅ then
                    Pi := Pi-1∪{ki};   SCP(i+1);
                end if;
            end for;

            ki := 0;    yet-to-be-covered := yet-to-be-covered ∪ {i};
            Pi := Pi-1;   SCP(i+1);
            yet-to-be-covered := yet-to-be-covered \ {i};
                                              { restore previous value }
        end if;
    else { i=m+1 }
        for all i'∈yet-to-be-covered do
            if |Pm∩Ki'| < 2 then
                {  i' is not covered at least twice in  ∪ Mj  }
                                                       j∈P
                return from procedure;
            end if;
        end for;
        print (Pm);
    end if;
end SCP;
```

Before calling SCP(1) initially, the sets yet-to-be-covered and P_0 must be set to $\emptyset$.

If the linearly ordered group B is bounded from below by some element z, then the procedure produces all minimal solutions of (1). Setting for each $P_m \subseteq \{1,\dots,n\}$ which is reported

$$x_j = \begin{cases} c_j, & \text{if } j \in P_m \\ z, & \text{if } j \notin P_m \end{cases} \quad \text{for all } 1 \leq j \leq n,$$

one obtains a list of the minimal solutions.

The largest solution is the vector $(c_1,\dots,c_n)$, if the solution set is not empty.

(Analogous results are of course obtained for the minsum problem with a linearly ordered group bounded from above, the set of maximal solutions, and the smallest solution.)

Since the set of minimal solutions correponds one-to-one to the set of minimal coverings, and the latter set is a set of pairwise incomparable subsets of $\{1,\dots,n\}$ (an anti-chain), it follows by standard combinatorial results that the number of minimal solutions is not greater than

$$\binom{n}{\lfloor n/2 \rfloor} \approx 2^n \cdot \sqrt{\frac{2}{n \cdot \pi}}$$

This bound can however only be attained tightly if the minimal coverings are either all sets with $\lfloor n/2 \rfloor$ or all sets with $\lceil n/2 \rceil$ elements, i. e. if m is equal to the above number or greater.

5. Special case (b) Maxmin (minmax, resp.) equations

Here we have $X_1=\ldots=X_n=B_1=\ldots=B_m=:B$, where $(B,\leq)$ is a linearly ordered set, and in (1) $f_{ij}(x) = \min(a_{ij}, x)$ for given elements $a_{ij}\in B$.
If we specialize the methods of the preceding sections for this problem we obtain

$$u_{ij} = \{ x\in B \mid \min(a_{ij},x)\leq b_i \} = \begin{cases} \{ x\in B \mid x\leq b_i \}, & \text{if } a_{ij}>b_i \\ B, & \text{if } a_{ij}\leq b_i \end{cases}$$

and

$$l_{ij} = \{ x\in B \mid \min(a_{ij},x)<b_i \} = \begin{cases} \{ x\in B \mid x<b_i \}, & \text{if } a_{ij}\geq b_i \\ B, & \text{if } a_{ij}<b_i \end{cases}$$

Thus, for each i, there are only two possibilities for the sets l_{ij}. For the sequences $\underline{p}$ with non-empty $X_{\underline{p}}$ (see the remark following the characterizations 3 and 3') the possibility $l_{ij}=B$ can be excluded, since then we can never have $l_{ij}\subset u_j$. Thus, with

$$P_j = \{ p \mid l_{pj}\subset u_j \},$$

we can conclude: $i\in P_j \Rightarrow l_{ij} = \{ x\in B \mid x<b_i \}$.

Therefore, in the generalized set covering problem referred to in characterization 4 we have:

$$\underline{M}^{(j)} = \{\emptyset\} \cup \{ M_i^{(j)} \mid i\in P_j \}, \text{ with}$$

$$\begin{aligned} M_i^{(j)} &= \{ i' \mid 1\leq i'\leq m,\ l_{i'j}\subseteq l_{ij} \} = \\ &= \{ i' \mid 1\leq i'\leq m,\ i'\in P_j,\ b_{i'}\leq b_i \} = \\ &= \{ i' \mid 1\leq i'\leq m,\ b_{i'}\leq b_i \} \cap P_j, \end{aligned}$$

The first set in this last intersection is independent of j, and the second set depends only on j.

Example 4:

Let B be the set $\mathbb{R}^+$ of non-negative real numbers, let m=6, n=5, and let the matrix A and the vector b be defined as follows:

$$(a_{ij}) = \begin{pmatrix} 12 & 10 & 20 & 20 & 10 \\ 8 & 1 & 8 & 3 & 10 \\ 8 & 8 & 7 & 6 & 12 \\ 1 & 1 & 3 & 3 & 1 \\ 2 & 2 & 3 & 3 & 3 \\ 1 & 1 & 3 & 1 & 3 \end{pmatrix}, \quad b_{(i)} = \begin{pmatrix} 10 \\ 8 \\ 8 \\ 3 \\ 3 \\ 3 \end{pmatrix}.$$

The upper bounds on the sets u_j are then

$$u_j: \quad 10 \quad \infty \quad 10 \quad 10 \quad 8$$

and we get the following data for the set covering problem:

i: | j:	1	2	3	4	5
1:	2	2	3	2	-
2:	1	-	2	-	2
3:	1	1	-	-	2
4:	-	-	1	1	-
5:	-	-	1	1	1
6:	-	-	1	-	1

The lines in the table form three groups: line 1; lines 2 and 3; and lines 4, 5, and 6, corresponding to the three values of the right sides b_i.

In the procedure GSCP2, the order of processing of the indices i in the hierarchy of recursive calls is arbitrary. If they are processed in order of non-increasing b_i, then an anticipation of the final test, whether a minimal covering has indeed been produced, is possible:

Assume that the procedure is at some instant called at level i, and all indices i' with $b_{i'}>b_i$ have been processed, i. e. those $k_{i'}$ are fixed. The procedure will now select some k_i and some set $M_{p_{k_i}}(k_i)$ which is the smallest set of the form $M_p(k_i)$ that covers the element i. If there is such a set at all, i. e. if $i\in P_{k_i}$ then

$$M_{p_{k_i}}(k_i) = \{ 1\leq i'\leq m \mid b_{i'}\leq b_i \} \cap P_{k_i} \subseteq \{ 1\leq i'\leq m \mid b_{i'}\leq b_i \},$$

i. e. no i' with $b_{i'}>b_i$ can become covered at this level, nor at any lower level. Therefore the test for covering which is normally executed at the end for each i'εyet-to-be-covered can already be carried out at this level if $b_{i'}>b_i$.

We can thus conclude that if the indices i are processed in order of non-increasing b_i, then each block of equal b_i can be processed completely before control is passed to the level of the next lower b_i. It turns out that the processing of one block of equal b_i involves just one set covering problem. These facts are reflected in the following procedure, which looks very short and elegant but uses a subroutine for enumerating all minimal solutions of a set covering problem. A specialized version of procedure GSCP1 or procedure SCP can be used here.

Let's assume that the set $\{b_1,\ldots,b_m\}$ contains r different values: $b_{(1)}>b_{(2)}>\cdots>b_{(s)}>\cdots>b_{(r)}$, and define

$$I_{(s)} := \{ i \mid b_i=b_{(s)} \}.$$

```
procedure MAXMIN(s);
  { At this level, the procedure will select the sets which cover I(s). }

    I' := { i∈I(s) |  i ∉  ∪    P_j };  { the set of uncovered elements }
                          p_j>0
    J' := { j | p_j=0 };

    for all minimal solutions J(s)⊆J' of the set covering problem
                            ∪   P_j  ⊇  I'
                          j∈J(s)
    do
        for all j∈J(s) do  p_j := s; end for;
        if s<r then        MAXMIN(s+1);
        else               print (p_1,...,p_n);
        end if;
        for all j∈J(s) do  p_j := 0; end for;  { reset to previous value. }
    end for;
end MAXMIN;
```

$p_1,\ldots,p_n$ must be initialized to 0 before the call to MAXMIN(1).

If the set I' is empty then there is only one minimal covering: $J_{(s)}=\emptyset$, and control passes straight through to the next level.
The condition $J_{(s)}\subseteq J'$ is not necessary for the definition of the set covering problem but helps only to make it smaller, because a set P_j with $j\notin J'$ could not cover any element in I', since the elements which might have been covered by P_j are already removed from I'.

The variables p_j are to be understood in such a way that their values now mean the parenthesized ordered indices of b:

$$X_{\underline{p}} = \{ (x_1,\ldots,x_n) \mid \text{for all } 1\leq j\leq n\text{: } x_j\geq b_{(p_j)} \text{ if } p_j>0\text{; and } x_j\in u_j \}.$$

Figure 5 shows the tree of calls of procedure MAXMIN for example 4.

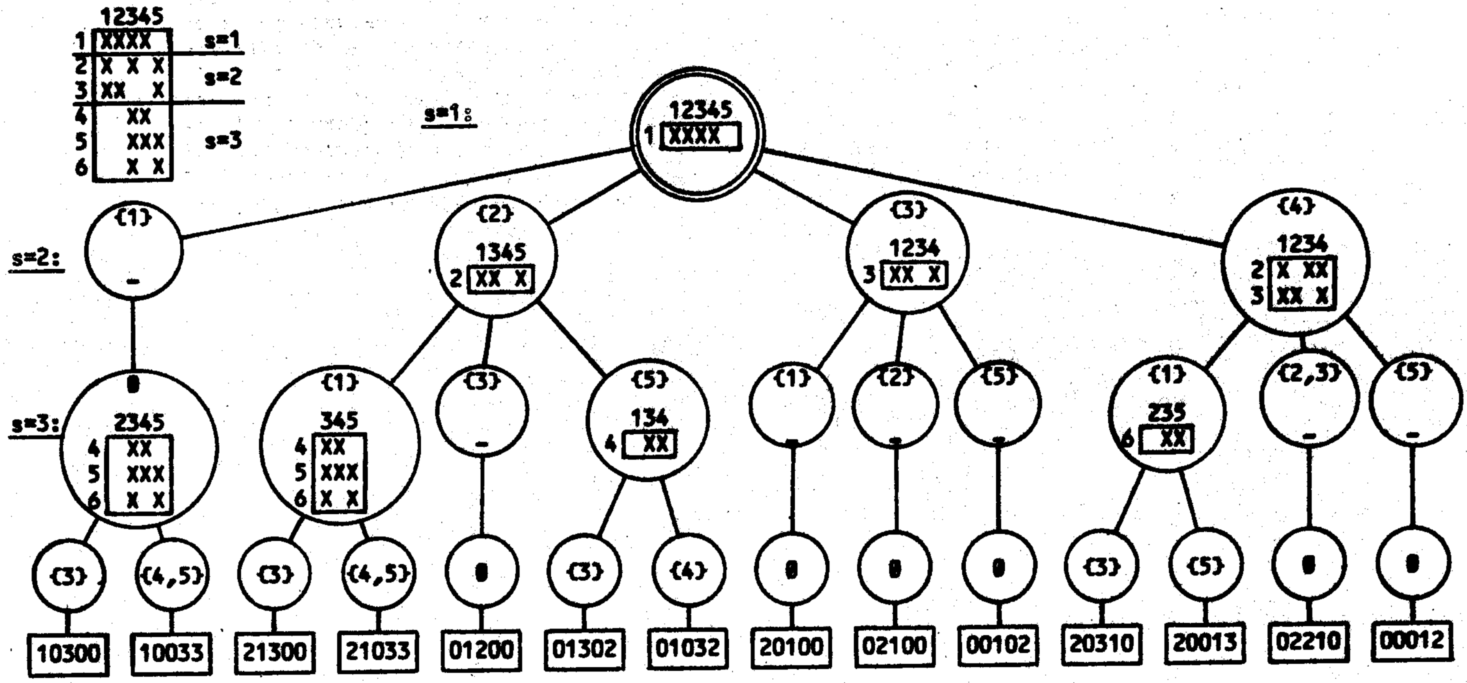

Figure 5 - tree of calls of the procedure MAXMIN for example 4

Each node corresponds to a call of the procedure MAXMIN; in each node the local set covering problem whose minimal solutions have to be determined is given: The rows are labeled by the elements of I', and the columns are labeled by the elements of J'. An entry X in some place in the matrix indicates that the element which corresponds to the row is contained in the set P_j corresponding to the column. If the set I' is empty this has been indicated as "-". The children of each node are labeled by the minimal covering solution to which they correspond. In the last row, the sequences $(p_1,..,p_5)$ which are produced by the program are given.
The data of the problem is repeated in the upper left corner, but only the sets P_j and the division of the rows into groups with equal right-hand side b_i is given.

If the set B is bounded from below by some element z, then the procedure will report the minimal solutions of (1) as the elements $(b_{(p_1)},\dots,b_{(p_n)})$, with the convention that $b_{(0)}=z$.

When $|I_{(s)}|=1$, then $|I'|\leq 1$ and the set covering problem becomes trivial. The procedure can then be coded in the following way, which is similar to procedure SCP. The sets

$$K_i = \{ j \mid i\in P_j \} = \{ j \mid l_{ij} \subset u_j \}$$

are again used.

```
procedure MAXMIN1(s);
   { |I(s)|=1 and I(s)={i}. }
   { At this level, the procedure will select the set which covers i. }

     if Ki ∩ { j | pj>0 } ≠ ∅ then
         if s<r then      MAXMIN1(s+1);
         else             print (p1,...,pn);
         end if;
     else
         for all ki∈Ki do
             pki := s;
             if s<r then      MAXMIN1(s+1);
             else             print (p1,...,pn);
             end if;
             pki := 0;  { reset to previous value. }
         end for;
     end if;
end MAXMIN1;
```

The maximal number of minimal solutions of maxmin equations

Let $N(m,n)$ be the maximal number of minimal solutions that a system of maxmin equations of size $m \times n$ may have. The above algorithm can be use to estimate this number. For the purpose of deriving an upper bound, I assume w. l. o. g. that all b_i are different and that $b_1 > b_2 > \cdots > b_m$.
Note that the behavior of the algorithm and hence also the number of minimal elements reported is completely determined by the sets K_i. Let's call this number $N(K_1,\ldots,K_m)$, then $N(m,n)$ is defined as

$$N(m,n) = \max_{\substack{\text{for all } i: \\ K_i \subseteq \{1,\ldots,n\}}} N(K_1,\ldots,K_m).$$

At each call of the procedure there are two possibilities, corresponding to the outermost alternation in the procedure: either the procedure passes control straight through to the next level, or it tries all possibilities in K_i. Suppose that $(K_1,\ldots,K_m)$ is a family of subsets of $\{1,\ldots,n\}$. Note that $\{j \mid p_j > 0\} = \{k_1,k_2,\ldots,k_{i-1}\} \setminus \{0\}$ at level i. The last level in the hierarchy of recursive procedure calls is the level m. The number of calls to this level is equal to $N(K_1,\ldots,K_{m-1})$. Lets assume that the number of calls where $K_m \cap \{j \mid p_j > 0\}$ is empty is a. Then in these calls the procedure will output $|K_m|$ minimal solutions, and in the remaining calls it will output one minimal solution. Thus we have

$$\begin{aligned} N(K_1,\ldots,K_m) &= a \cdot |K_m| + (N(K_1,\ldots,K_{m-1}) - a) \cdot 1 = \\ &= N(K_1,\ldots,K_{m-1}) + a \cdot (|K_m| - 1) \ . \end{aligned}$$

The number a is the number of possibilities that no element of K_m has been selected in the first m-1 steps. It can be written as follows:

$$a = N(K_1 \backslash K_m, \ldots, K_{m-1} \backslash K_m) \leqq N(m-1, n-|K_m|) \;.$$

Here $N(m-1,0)$ is of course 0, unless $m-1=0$.

Putting these results together gives:

$$N(K_1,\ldots,K_m) \leqq N(K_1,\ldots,K_{m-1}) + N(m-1,n-|K_m|)\cdot(|K_m|-1).$$

Using induction on m, we obtain:

$$N(K_1,\ldots,K_m) \leqq N(K_1) + \sum_{2\leqq i\leqq m} N(i-1,n-|K_i|)\cdot(|K_i|-1),$$

or, since $N(K_1)=|K_1|$,

$$\text{(10)} \quad N(K_1,\ldots,K_m) \leqq 1 + \sum_{1\leqq i\leqq m} N(i-1,n-|K_i|)\cdot(|K_i|-1),$$

if we set $N(0,n)=1$, for all n.

By taking the maximum over all $(K_1,\ldots,K_m)$ with $K_i \subseteq \{1,\ldots,n\}$, and setting $d_i := |K_i|$:

$$\text{(11)} \quad N(m,n) \leqq 1 + \max_{\substack{(d_1,d_2,\ldots,d_m) \\ 1\leqq d_i\leqq n}} \sum_{1\leqq i\leqq m} N(i-1,n-d_i)\cdot(d_i-1),$$

This bound can still be enhanced, by the following observation. If two sets $K_{i'}$ and K_i, $i'<i$, are equal (or more generally, if $K_{i'} \subseteq K_i$), then

$$N(K_1,\ldots,K_{i'},\ldots,K_{i-1},K_i,K_{i+1},\ldots,K_m) =$$
$$= N(K_1,\ldots,K_{i'},\ldots,K_{i-1},K_{i+1},\ldots,K_m) =$$
$$= N(K_1,\ldots,K_{i'},\ldots,K_{i-1},K_{i+1},\ldots,K_m,\{1\}).$$

For if $K_{i'}=K_i$ then at level i the case $K_i \cap \{j|p_j>0\} \neq \emptyset$ can never occur, because some p_j with $j \in K_{i'} \subseteq K_i$ has been set to i' at level i'. Therefore nothing changes in the number of minimal elements produced, if level i is omitted altogether. Adding a one-element set at the end again changes nothing.

Therefore we can always assume without loss of generality that there are no equal sets in $(K_1,\ldots K_m)$ except sets of cardinality 1.

In the transition from inequality (10) to inequality (11) for $N(m,n)=\max(N(K_1,\ldots K_m))$ it suffices to take the maximum over all sequences $(K_1,\ldots K_m)$ with no equal sets except sets with only one element. Since there are $\binom{n}{k}$ different subsets of $\{1,\ldots,n\}$ of cardinality k, and since the sequence $(d_1,\ldots,d_m)$ represents the cardinalities of the sets $K_1,\ldots,K_m$, we may impose the following additional restriction on the sequence $(d_1,\ldots d_m)$ in formula (11):

(12) for all $k \geq 2$: $$|\{i|d_i=k\}| \leq \binom{n}{k} .$$

The bounds on $N(m,n)$ from formula (11) can easily be computed recursively, if $N(m-1,n)$ and all $N(m',n')$ with $m'<m$ and $n'<n$ are known.
The problem of determining the bound $N(m,n)$ subject to the conditions (11) and (12) can be formulated as a transportation problem:

> The set of sources is $\{1,\ldots,n\}$, with a supply of $\binom{n}{k}$ units at source k, for $k\geq 2$, but with unlimited supply at source 1. There are m sinks, numbered from 1 to m, each with unit demand. The cost of sending one unit from source k to sink i is $N(i-1,n-k)\cdot(k-1)$, and the total costs of a solution is the sum of the unit costs. We are looking for a maximum cost solution satisfying all demands.

If the shortest augmenting path method is used to solve the transportation problem, then the problem for $N(m,n)$ can be solved starting from the problem for $N(m-1,n)$ using just one augmenting path, since the costs are independent of m and each sink has a demand of one unit, and therefore the maximum flow after adding sink m increases by 1.

The values obtained in this way are shown in the tables 1 and 2. The entries of table 1 are the correct values of $N(m,n)$ for $n\leq 4$. The first bound which is not tight is the bound for $N(6,5)=29$, which can be proved by enumerative techniques and case distinctions.
Table 3 disproves a conjecture of Czogala, E., J. Drewniak, and W. Pedrycz [1982, at the end of section 3], that $N(n,n)<2^n$.

Table 1: Upper bounds for N(m,n) from (11) and (12)

m: | n:	1	2	3	4	5	6	7	8	9	10	11
1:	1	2	3	4	5	6	7	8	9	10	11
2:	1	2	4	6	9	12	16	20	25	30	36
3:	1	2	5	8	13	20	28	38	52	66	84
4:	1	2	6	10	18	30	44	64	92	126	168
5:	1	2	6	12	24	42	64	100	152	216	300
6:	1	2	6	14	30	54	88	148	236	344	500
7:	1	2	6	16	36	68	118	208	344	520	796
8:	1	2	6	18	42	84	154	280	480	756	1212
9:	1	2	6	20	48	102	196	364	648	1064	1772
10:	1	2	6	22	54	122	244	466	852	1456	2500
11:	1	2	6	24	60	144	298	588	1096	1944	3432
12:	1	2	6	24	66	168	358	732	1394	2540	4608
13:	1	2	6	24	72	192	424	900	1752	3272	6072
14:	1	2	6	24	77	216	496	1092	2176	4172	7872
15:	1	2	6	24	81	240	573	1308	2672	5264	10056

Table 2: Upper bounds for N(n,n) from equations (11) and (12)

n	
1	1
2	2
3	5
4	10
5	24
6	54
7	118
8	280
9	648
10	1456
11	3432
12	8028
13	18285
14	43206
15	101433
16	232788
17	550825
18	1296372
19	2990840
20	7083052
21	16701918
22	38672898
23	91655159
24	216458550
25	502530194
26	1191761402
27	2817844839
28	6555357096
29	15555628432
30	36812236936
31	85780683920
32	203664882290
33	482305473371
34	1125422750160
35	2673230788463
36	6334226103548
37	14797659376246
38	35162131303894

Table 3: An example exhibiting that N(8,8)≥258

i	K_i
1	{ 1,2,3,4,5,6,7,8 }
2	{ 1,2,3,4 }
3	{ 4,5,6 }
4	{ 1, 6,7 }
5	{ 2, 6, 8 }
6	{ 3, 7,8 }
7	{ 2, 5, 7 }
8	{ 1, 5, 8 }

An upper bound independent of m, which might be guessed by looking at the first four columns of table 1, can be proved to be tight for large m:

$N(m,n) \leq n!$;

furthermore, $N(m,n)=n!$ if and only if $m \geq 2^n-n-1$.

__Proof:__

__Step 1:__ We start by proving that $m \geq 2^n-n-1$ implies $N(m,n) \geq n!$.
Let $m=2^n-n-1$ and take $(K_1,K_2,\dots,K_m)$ to be the sequence consisting of all different subsets of $\{1,2,\dots,n\}$ with at least two elements in the following order: $K_1=\{1,\dots,n\}$; next come all subsets of cardinality n-1 in arbitrary order, then all subsets of cardinality n-2 in arbitrary order, and so on; the subsets of cardinality 2 are last. To complete step 1 of the proof we shall show that for each permutation $(q_1,q_2,\dots,q_n)$ of the set $\{1,\dots,n\}$ there is a sequence $(k_1,\dots,k_m)$ by which the algorithm MAXMIN1 produces a minimal covering, and $(k_1,\dots,k_m)$ coincides with the sequence $(q_1,\dots,q_{n-1})$ when the elements k_i with $k_i=0$ are removed.
This can be seen as follows:
At level 1 the procedure can choose $k_1=q_1$ to be any of the n elements in K_1. Among the sets K_i with cardinality n-1 there is exactly one set K_{i_2} which is disjoint from $\{k_1\}=\{q_1\}$ such that the second alternative is chosen in the procedure. (In all other cases k_i is set to 0.) $k_{i_2}=q_2$ can then be chosen as any number different from q_1. Among the sets K_i with cardinality n-2, which come next, there is exactly one set K_{i_3} which is disjoint from $\{q_1,q_2\}$, and $k_{i_3}=q_3$ can be chosen to be any number different from q_1 and q_2, and so on.

<u>Step 2:</u> The inequality $N(m,n)\leqq n!$ can be derived from formula (11) with the restriction (12) by induction on n.

For n=1, it is certainly true.

Combining terms with equal d_i in the sum (11), and inductively replacing $N(i-1,n-k)$ by the bound $(n-k)!$, yields

$$N(m,n) \leqq 1 + \max_{\substack{(d_1,d_2,\dots,d_m)\\ 1\leqq d_i\leqq n}} \sum_{1\leqq k\leqq n} \sum_{\substack{1\leqq i\leqq m\\ d_i=k}} N(i-1,n-k)\cdot(k-1) \leqq$$

$$\leqq 1 + \max_{\substack{(d_1,d_2,\dots,d_m)\\ 1\leqq d_i\leqq n}} \sum_{1\leqq k\leqq n} \sum_{\substack{1\leqq i\leqq m\\ d_i=k}} (n-k)!\cdot(k-1) \leqq$$

(13) $$\leqq 1 + \max_{\substack{(d_1,d_2,\dots,d_m)\\ 1\leqq d_i\leqq n}} \sum_{1\leqq k\leqq n} |\{i|d_i=k\}|\cdot(n-k)!\cdot(k-1) .$$

For k=1, the term under the rightmost sum does not contribute anything, and for $k\geqq 2$, restriction (12) can be applied:

(14) $$N(m,n) \leqq 1 + \sum_{1\leqq k\leqq n} \binom{n}{k}\cdot(n-k)!\cdot(k-1) = 1 + n! \cdot \sum_{1\leqq k\leqq n} \frac{k-1}{k!} .$$

By induction, this last sum can easily be proved to be $1-1/n!$, which yields the desired inequality.

In the transition from (13) to (14), equality can of course only be achieved if equality holds in (12). For this it is necessary that

$$m = \sum_{1\leqq k\leqq n} |\{i|1\leqq i\leqq m, d_i=k\}| \geqq \sum_{2\leqq k\leqq n} \binom{n}{k} = 2^n-n-1 .$$

Therefore, if $m<2^n-n-1$ then $N(m,n)$ is less then n!.

6. Special case (c) The set covering problem

Here we have $X_1=\ldots=X_n=B_1=\ldots=B_m=\{0,1\}$, where $(\{0,1\},\leq)$ is the two-element Boolean algebra $(\{0,1\},\vee,\wedge,0,1)$, where $\vee$ is written for max and $\wedge$ for min, and in (1) $f_{ij}(x) = \min(a_{ij}, x)$ for given elements $a_{ij}\in\{0,1\}$.

If $a_{ij}=0$ then $f_{ij}(x)=0$;
if $a_{ij}=1$ then $f_{ij}(x)=x$.
Hence, with $K_i:=\{j|a_{ij}=1\}$, we can write the system (1) as:

$$\max_{j\in K_i} x_j = b_i, \quad \text{for } i=1,2,\ldots,m,$$

If $b_i=0$, then the corresponding equation is equivalent to the condition that all x_j with $j\in K_i$ are 0. If all the equations with $b_i=0$ are applied in this way and the corresponding variables are eliminated from the remaining equations, one gets, after appropriate renumbering, a system of equations of the form:

$$\max_{j\in K_i} x_j = 1, \quad \text{for } i=1,2,\ldots,m,$$

Clearly, this is a set covering problem: we are looking for a subset of the sets $\{P_j|1\leq j\leq n\}$, where $P_j=\{i|j\in K_i\}$, which covers the set $\{1,2,\ldots,m\}$. A solution $(x_1,\ldots,x_n)$ of the above system corresponds to a covering $\{P_j|x_j=1\}$.

The set covering problem has been discussed in the previous section.

Systems of set equations of the form

$$\bigcup_{1\leq j\leq n} (a_{ij} \cap x_j) = b_i, \quad \text{for } i=1,2,\ldots,m,$$

where a_{ij}, b_i, and the unknowns x_j are subsets of a given common ground set S, can be decomposed into unrelated systems of type (1') over the two element Boolean algebra, by regarding each element of the set S separately. More generally, if the algebraic structure (B,∗) is the direct product of the linearly ordered structures $(B_s,*_s)$, s∈S, for some index set S, then each equation of the system (1') over (B,∗) is equivalent to a system of |S| unrelated equations of the same form over the structures $(B_s,*_s)$, since the max and ∗ operations as well as the = sign can be taken componentwise. Thus, the system (1') is equivalent to |S| unrelated systems of the form (1'), one system for each s∈S. In the case mentioned above, the set algebra of the subsets of a ground set S, (P(S),∪,∩), is the direct product of |S| copies of the Boolean algebra ({0,1},∨,∧). (P(S) denotes the power-set of S.)

7. Conclusion

All the general results in section 2 carry over to systems with antitone functions f_{ij} or to systems with the maximum operation replaced by the minimum operation; by inverting the linear order on the sets $X_1, X_2, \dots, X_n$ or on the sets $B_1, B_2, \dots, B_m$, such systems can be transformed into equivalent systems of the original type.

In section 2, the equalities were treated by splitting them into two inequalities with "$\leq$" and "$\geq$" and treating these inequalities separately; hence the methods and results of section 2 can also be applied to systems where the equality sign is replaced by "$\leq$" or "$\geq$" in some or all of the equations of the system (1). Furthermore, the methods can be extended to handle also strict inequalities ("<" and ">") without difficulty.

In all the given examples and in all applications that are reported in the literature, the functions $f_{ij}(x)$ can be written in the form $a_{ij} * x$ in some algebraic structure. It is important to stress that the general results are not algebraic in nature, but order-theoretic: they rely only on the linear ordering of the ground set and on the monotonicity of the functions f_{ij}. Partially ordered sets B_i which are direct products of linearly ordered sets can be handled like in the last part of section 6. If the isotonicity condition of the functions f_{ij} is relieved, then the structure of the solution set becomes very complicated, since the crucial property that the sets $u_j \setminus l_{ij}$ form a chain (for fixed j) does not hold any more.

Finally we remark that in order to minimize a function of the form

$$\max_{1 \leq j \leq n} c_j(x_j) \quad \text{or} \quad \min_{1 \leq j \leq n} c_j(x_j)$$

with isotone functions $c_j : X_j \to C$ (C is a linearly ordered set) over the set of feasible solutions described by the system (1), it is not necessary to enumerate the minimal feasible solutions (cf. U. Zimmermann [1979, 1981], K. Zimmermann [1982]). However, for minimizing other isotone functions $c(x_1, \dots, x_n)$, enumerating the minimal feasible solutions might be useful.

References

Cuninghame-Green, R. [1979]
Minimax Algebra.
Lecture Notes in Economics and Mathematical Systems 166
Springer-Verlag; Berlin, Heidelberg, New York 1979

Czogala, E., J. Drewniak, and W. Pedrycz [1982]
Fuzzy relation equations on a finite set.
Fuzzy Sets and Systems 7 (1982), 89-101

Garey, M. R., and D. S. Johnson [1979]
Computers and Intractability, A Guide to the Theory of NP-Completeness
W. H. Freeman and Company, San Francisco 1979

Garfinkel, R. S., and G. L. Nemhauser [1972]
Integer Programming.
John Wiley & Sons; New York etc. 1972

Hammer (Ivănescu), P. L., and S. Rudeanu [1968]
Boolean Methods in Operations Research and Related Areas.
Econometrics and Operations Research VII
Springer-Verlag; Berlin, Heidelberg, New York 1968

Lawler, E. L., J. K. Lenstra, and A. H. G. Rinnooy Kan [1980]
Generating all maximal independent sets: NP-hardness and polynomial-time algorithms.
SIAM J. Computing 9 (1980), 558-565

Sanchez, E. [1976]
Resolution of Composite Fuzzy Relation Equations.
Information and Control 30 (1976), 38-48

Sysło, M. M., N. Deo, and J. S. Kowalik [1983]
Discrete Optimization Algorithms with PASCAL Programs
Prentice-Hall, Inc.; Englewood Cliffs, New Jersey 1983

Vorob'ev, N. N. [1963]
The extremal matrix algebra. (Russian)
Doklady Akademii Nauk SSSR, Tom 152 (1963), no. 1, 24-27
(English translation in:
Soviet Mathematics Doklady 4 (1963), 1220-1223)

Vorob'ev, N. N. [1967]
The extremal algebra of positive matrices. (Russian)
EIK (Elektronische Informationsverarbeitung und Kybernetik) 3
(1967), 39-71

Zimmermann, K. [1973-a]
Some properties of systems of extremal linear inequalities.
(Russian)
Ekonomicko-matematický obzor 9 (1973), 212-227

Zimmermann, K. [1973-b]
Solution of some optimization problems in the extremal vector
space. (Russian)
Ekonomicko-matematický obzor 9 (1973), 336-351

Zimmermann, K. [1974-a]
Solution of one optimization problem in the extremal vector space.
(Russian)
Ekonomicko-matematický obzor 10 (1974), 298-321

Zimmermann, K. [1974-b]
Conjugate optimization problems and algorithms in the extremal
vector space.
Ekonomicko-matematický obzor 10 (1974), 428-440

Zimmermann, K. [1976]
Extremální Algebra.
Výzkumná publikace Ekonomicko-matematické Laboratoře při
Ekonomickém Ústavu ČSAV, VP č. 46, Praha 1976

Zimmermann, K. [1982]
Explicit Solution of Max-Separable Optimization Problems.
Ekonomicko-matematický obzor 18 (1982), 442-454

Zimmermann, K. [1984]
On max-separable optimization problems.
in: R. E. Burkard, R. A. Cuninghame-Green, U. Zimmermann (eds.), Algebraic and Combinatorial Methods in Operations Research (Proceedings of the Workshop on Algebraic Structures in Operations Research, Bad Honnef, Federal Republic of Germany, April 13-17 1982), Ann. Discrete Math. 19 (1984), 357-362

Zimmermann, K., and F. Juhnke [1979]
Lineare Gleichungs- und Ungleichungssysteme auf extremalen Algebren.
Wissenschaftliche Zeitschrift der Technischen Hochschule Otto von Guericke Magdeburg 23 (1979), 103-106

Zimmermann, U. [1979]
On some extremal optimization problems.
Ekonomicko-matematický obzor 15 (1979), 438-442

Zimmermann, U. [1981]
Linear and Combinatorial Optimization in Ordered Algebraic Structures.
(especially chapter 10: Extremal Linear Programs)
Ann. Discrete Math. 10 (1981), 1-380