ON REDUNDANCY IN SYSTEMS
OF LINEAR INEQUALITIES

J. TELGEN

REPORT 7718/0

ERASMUS UNIVERSITY ROTTERDAM, P.O. BOX 1738, ROTTERDAM, THE NETHERLANDS
ON REDUNDANCY IN SYSTEMS OF LINEAR INEQUALITIES
by Jan Telgen

Abstract

In this paper the concept of redundancy in systems of linear inequalities is established from the existence of the minimal inequality representation of a system of linear constraints. It is shown that absence of redundancy is a necessary and sufficient condition for having a minimal inequality representation of the system; then a minimal inequality representation can be obtained by deleting redundant constraints.

Furthermore a general method to determine redundancy is developed; this method is based on the simplex method and is greatly inspired by Gal [2]. A number of known methods can be shown to be simplified variants of this method.

Finally the equivalence, in terms of complexity theory, of the problem of determining redundancy and the general linear programming problem, is proved. From this a class of problems is indicated, for which it may be fruitful to determine redundancy.

Contents

1. Introduction 2
2. Minimal inequality representation 3
3. Redundancy 3
4. A general method to determine redundancy 6
5. Variants of the general method 9
6. Redundancy and complexity 10
Acknowledgment 12
References 13

september 1977
1. INTRODUCTION

We consider the system of linear inequalities

\[ \sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad i = 1, \ldots, m \]

which may alternatively be written

\[ Ax < b \]

with \( A, b \) and \( x \) of dimensions \((m \times n), (m \times 1)\) and \((n \times 1)\), respectively.

The feasible region or the set of all feasible solutions to (1.1) is defined by:

\[ S = \left\{ x \in \mathbb{R}^n \mid Ax \leq b \right\} \]

It is assumed that \( S \neq \emptyset \); in other words, there exists at least one feasible solution to (1.1).

In section 2 the existence of a minimal inequality representation for the set \( S \) is indicated. This means that a minimal value \( m^* \) exists for the number of constraints that describe the set \( S \).

From this fact the concept of redundancy is established in section 3. Then it is proved that one general criterion can be used to determine whether a system is in its minimal inequality representation or not. This same criterion can be used to determine redundancy.

In section 4 from this criterion a general method to determine redundancy is developed based on the simplex method and greatly influenced by the work of Gal [2]. In section 5 it is shown that some known methods to identify redundant constraints are in fact variants of this general method. Section 6 is devoted to the relation between redundancy and linear programming. It is shown that these problems are equivalent from a complexity point of view. Then a number of problems are indicated for which it may be fruitful to determine redundancy, since they are more complex than general linear programming.
2. MINIMAL INEQUALITY REPRESENTATION

First observe that any system of linear constraints can be denoted in the form (1.1);

equality constraints

\[ (2.1) \sum_{j=1}^{n} a_{ij}x_j = b_i \]

can be written as two inequality constraints:

\[ (2.2) \begin{cases} \sum_{j=1}^{n} a_{ij}x_j \leq b_i \\ -\sum_{j=1}^{n} a_{ij}x_j \leq -b_i \end{cases} \]

constraints in \( \geq \) form can be multiplied by \(-1\) to obtain the \( \leq \) form. \(^1\)

The set \( S \) of all feasible solutions may be represented in a number of ways. We define a minimal inequality representation of the set \( S \) to be a system of inequalities

\[ (2.3) \sum_{j=1}^{n} a_{ij}x_j \leq b_i \quad i = 1, \ldots, m^* \]

Such that

\[ (2.4) S = S^* = \{ x \in \mathbb{R}^n \mid \sum_{j=1}^{n} a_{ij}x_j \leq b_i \quad i = 1, \ldots, m^* \} \]

while \( m^* \) is on its minimal value.

Some authors, e.g. Luenberger [5] refer to a minimal representation of the set \( S \) meaning a minimal inequality representation using a minimal number \( \sigma (\leq n) \) of variables and adding \( n-\sigma \) equality constraints. To avoid any confusion we prefer to use the word minimal inequality representation.

3. REDUNDANCY

The redundancy \( R \) of a system of linear inequalities is defined as the difference between the number of inequalities \( m \) and the minimal

\(^1\)The same argument applies to fixed variables, e.g. \( x_j = d \) can be written as \( x_j \leq d \) and \( -x_j \leq -d \).
number of inequalities $m^*$

\[(3.1)\quad R = m - m^*\]

If $R \geq 1$ the system of linear inequalities can be replaced by another system, with less inequalities, but describing the same set of solutions $S$. If $R = 0$, the system is in its minimal inequality representation. Now it is very important that this smaller system and consequently a minimal inequality representation can be obtained by deleting certain inequalities from the original system according to a suitable redundancy criterion:

**Definition:** An inequality constraint in (1.1), say the $k$-th, is redundant if and only if

\[(3.2)\quad \min_{S_k} u_k = \min_{S_k} (b_k - \sum_{j=1}^{n} a_{kj}x_j) \geq 0\]

where $S_k = \{x \in \mathbb{R}^n \mid \sum_{j=1}^{n} a_{ij}x_j \leq b_i \quad i = 1, \ldots, m, i \neq k\}$

In this definition $k$ may range from 1 to $m$, where it is assumed that all constraints are in "less than or equal to" -form. For nonnegativity constraints on a variable $x_1$ this means that (3.2) can be replaced by:

\[(3.3)\quad \min_{S_k} u_k = \min_{S_k} (0 - (-x_1)) = \min_{S_k} x_1 \geq 0\]

The following theorem is essential:

**Theorem:** A system of linear inequalities is in a minimal inequality representation (I) if and only if (II) for all $k \leq m$

$$\min_{S_k} u_k < 0$$

holds.

**Proof:** (I) $\Rightarrow$ (II)

Assume (II) does not hold, i.e. there is at least one inequality $k' \leq m$ with $\min_{S_{k'}} u_{k'} \geq 0$, or

\[(3.4)\quad u_{k'} = b_{k'} - \sum_{j=1}^{n} a_{k'j}x_j \geq 0 \quad \forall x \in S_{k'}\]

\[(3.5)\quad \sum_{j=1}^{n} a_{k'j}x_j \leq b_{k'} \quad \forall x \in S_{k'}\]
so

\[(3.6) \quad x \in S \quad \forall x \in S_{k'} \]

and thus

\[(3.7) \quad S_{k'} \subseteq S \quad \text{which implies } S_{k'} = S \]

and this contradicts the fact that the system is in minimal inequality representation, since inequality \(k'\) can be deleted without altering the feasible region, if (II) does not hold. Therefore (II) should hold.

q.e.d.

(II) ⇒ (I):

From (II) we derive for any \(k' < m\)

\[(3.8) \quad \min_{S_{k'}} u_{k'} = \min_{S_{k'}} \left( b_{k'} - \sum_{j=1}^{n} a_{k',j} x_j \right) < 0 \]

then

\[(3.9) \quad \exists x \in S_{k'} \quad \sum_{j=1}^{n} a_{k',j} x_j > b_{k'} \]

\[(3.10) \quad \Rightarrow \exists x \in S_{k'} \quad \text{with } x \notin S \]

Therefore the \(k'\)-th constraint must be a face of the convex polyhedron formed by \(S\). Since this applies to all \(k < m\) constraints, the convex polyhedron formed by \(S\) must have \(m\) different faces.

Because all the inequalities are linear, all the faces of \(S\) are (part of) \((n-1)\) dimensional hyperplanes. Now since the convex polyhedron formed by \(S\) has \(m\) different linear faces, at least \(m\) different linear inequalities are required to describe \(S\). Then the minimal inequality representation of the system consists of \(m\) inequalities.

q.e.d.

For later use we will also define two kinds of redundant inequalities:

Definition: The \(k\)-th inequality constraint \((k < m+1)\) is strict redundant if

\[(3.11) \quad \min_{S_k} u_k > 0 \]

Definition: The \(k\)-th inequality constraint \((k < m+1)\) is weak redundant if

\[(3.12) \quad \min_{S_k} u_k = 0 \]

For a more detailed treatment of several definitions for redundant constraints and related concepts we refer to Telgen [6].
4. A GENERAL METHOD TO DETERMINE REDUNDANCY

In (3.2) we defined a constraint to be redundant if

\[ \min_{S_k} u_k \geq 0 \]

Therefore we would obtain a minimal inequality representation for a system by solving \( m \) linear programming problems (the left hand side of (4.1)) and deleting those constraints \( k \) for which (4.1) holds.

However, it is not entirely necessary to solve \( m \) distinct linear programming problems, if we use the following general method, which is strongly influenced by the work of Gal [2], and in fact arose from the discussions the author had with professor Gal. The general method is an extension of Gal's method because it can handle weak redundant inequalities as well as strict redundant constraints; furthermore some slight modifications are introduced.

It is assumed that a basic feasible solution to the system (1.1) is known and the corresponding simplex tableau is set up.

Following the notation of Gal [2] we define:

\[ I_1 = \text{the set of subscripts of the basic variables } x_j \ (1 \leq j \leq n) \]
\[ I_2 = \text{the set of subscripts of the basic slack variables } u_i \ (1 \leq i \leq m) \]
\[ N_1 = \text{the set of subscripts of the non-basic variables } x_j \ (1 \leq j \leq n) \]
\[ N_2 = \text{the set of subscripts of the non-basic slack variables } u_i \ (1 \leq i \leq m) \]

Note that for nonnegativity constraints the slack variables \( u_i \) are equal to the variables \( x_1 \) since by definition:

\[ u_i = 0 - (-x_1) = x_1 \]

Without loss of generality it is assumed that the subscripts \( j \in I_1 \) and \( i \in I_2 \) are numbered equally with the rows of the corresponding simplex tableau.
The method consists of the following steps:

**step 1: case (A):** the basic feasible solution is nondegenerate in the sign restricted basic variables. For all sign restricted basic variables \( y_k \) (where \( y_k \) is either \( u_k \) or \( x_k \)) we may write

\[
y_k = b_k - \sum_{j \in N1} a_{kj} x_j - \sum_{j \in N2} a_{kj} u_j
\]

with \( b_k > 0 \).

Since the solution is nondegenerate a slight reduction of \( u_j \) (\( j \in N2 \)) will not cause the solution to become infeasible. Therefore

\[
\min_{s_j} u_j < 0 \quad (j \in N2)
\]

and the corresponding constraints are not redundant.

**case (B):** the basic feasible solution is degenerate in the sign restricted basic variables. The argument used in case (A) does not apply in this case.

Now if in a column \( \alpha_j \) corresponding to a variable \( u_j \) with \( j \in N2 \) there is an element \( a_{kj} \) with sign opposite to the sign of \( b_k \), then (4.4) holds and the corresponding constraint is not redundant. This can be seen from the following argument:

Since there exists a row \( k \) with

\[
\frac{b_k}{a_{kj}} < 0 \quad j \in N2
\]

there also exists

\[
\frac{b_s}{a_{sj}} = \max_{k \in I, j \in I_2} \left( \frac{b_k}{a_{kj}} \right) \quad \frac{b_k}{a_{kj}} < 0
\]

Then \( a_{sj} \) can be used as a pivot to introduce \( u_j \) into the basis on a negative level, without causing other infeasibilities. Therefore (4.4) holds. Without actually performing a pivot iteration as indicated above, the constraints corresponding to \( u_j \) with \( j \in N2 \) for which (4.5) holds can also be determined as not redundant.

If condition (4.5) does not hold \( u_j \) should be pivoted into the basis; this is always possible either in one of the rows of the degenerate basic
variables or (if there are no nonzero elements in these rows) by using $a_{sj}$ as a pivot where

\[
\frac{b_s}{a_{sj}} = \min \left( \frac{b_{k1}}{a_{kj}}, \frac{b_{kj}}{a_{kj}} > 0 \right)
\]

Continue with step 2.

**step 2:** All rows in the tableau with $u_k, k \in I_2$ as a basic variable are considered for exhibiting the property:

\[
a_{kj} \leq 0 \quad j = 1, \ldots, m+n \quad j \neq k+n
\]

This property ensures that

\[
\min_{S \cup \{k\}} u_k = b_k
\]

since the $k$-th row can be considered as the criterion row for its basic variable $u_k$. For a formal proof we refer to Gal [2].

If (4.8) holds and $b_k > 0$ the corresponding constraint is strict redundant; if (4.8) holds and $b_k = 0$ the corresponding constraint is weak redundant.

Denote by $I_2'$ the set of subscripts $k \in I_2$ for which (4.8) holds.

**step 3:** In the criterion rows $i \in I_2 - I_2'$ for the remaining slack variables there exists at least one $a_{ij} > 0$ ($j \neq i + n$). Then, for all columns $a_j$ with

\[
a_{ij} > 0 \quad i \in I_2 - I_2' \quad j \neq i + n
\]

we determine

\[
\frac{b_s}{a_{sj}} = \min_{i \in I_2 - I_2'} \left( \frac{b_i}{a_{ij}} \mid a_{ij} > 0 \right)
\]

We distinguish between two cases: case (A): the minimum is unique. Then pivoting on $a_{sj}$ yields the situation described in case (A) of step 1, so without performing the actual iteration we may say that:
and consequently the corresponding constraint is not redundant.

case (B): the minimum is not unique. Performing a pivot operation
on \( a_{sj} \) would result in a degenerate solution, as described in
step 1, case (B).

If in column \( j \) there is an element \( a_{kj} > 0 \) with

\[
\frac{b_k}{a_{kj}} > \frac{b_s}{a_{sj}}
\]

then condition (4.5) would hold after an iteration with pivot \( a_{sj} \), so
without performing the iteration we determine the constraint corre-
sponding to \( u_s \) to be not redundant.

If (4.13) does not hold we continue with step 4.

\[ (4.12) \quad \min u_s < 0 \]

\( \begin{array}{c} \text{step 4: If for all slack variables } u_i \text{ the minimum has been determined,} \\ \text{then STOP. Otherwise continue with step 5.} \end{array} \]

\[ \text{step 5: All slack variables } u_i \text{ for which the minimum has not yet been} \]
\[ \text{determined are basic variables with at least one } a_{ij} > 0. \text{ Select} \]
\[ \text{one of these slack variables (e.g. the one with the smallest} \]
\[ \text{subscript } i \text{) and perform an iteration with pivot } a_{sj} \text{ as determined} \]
\[ \text{in (4.11). Go back to step 2.} \]

Note that it is possible to delete all rows corresponding to redundant
inequalities, as soon as they have been determined as such. This may reduce
the number of computations.

Nonnegativity constraints on the variables \( x_j, 1 \leq j \leq n \), can be treated
in the same way as the other constraints, by handling these variables in the
same way as the slack variables \( u_i \), according to (4.2).

5. VARIANTS OF THE GENERAL METHOD

As stated before the general method we developed is strongly influenced
and inspired by the work of Gal \([2]\) and the communications the author had with
professor Gal.

In the method of Gal \([2]\) no attention is paid to degenerate solutions
and therefore it is only useful to determine strict redundant inequalities. Furthermore, step 3 in our algorithm is formulated in a somewhat more general way and may cause fewer calculations to be done compared to Gal's method. Apart from these facts the methods are quite similar.

The method developed by Thompson, Tonge and Zionts [6] consists mainly of scanning each tableau for the situation as described in our step 2. No systematic way is indicated to obtain such a situation as is done in our steps 1, 3 and 5. Therefore the method of Thompson, Tonge and Zionts is merely an application of part of our step 2 to each simplex tableau that is generated.2

The rules given by Llewellyn [3] are a direct and an indirect application of the property we described in step 2 of our algorithm. However the rules as formulated by Llewellyn cannot be applied in all situations, as is possible with our algorithm. For a detailed critique and generalization we refer to Telgen [7].

Other methods, including heuristics, are described in Telgen [6]. However all these methods are merely rules to recognize specific situations which are part of (and in fact generated by) the general method.

6. REDUNDANCY AND COMPLEXITY

In section 5, it has been shown that redundant constraints can be identified by solving m linear programming problems. If we had an algorithm to solve linear programming problems in time bounded by a polynomial in problem size, then redundant constraints could be identified in polynomial time as well. The converse is also true: if a polynomially bounded algorithm to identify redundant inequalities were known, one should write the general LP-problem

$$\begin{align*}
\text{max} & \quad z = c^T x \\
\text{subject to} & \quad Ax \leq b \\
& \quad x \geq 0
\end{align*}$$

(6.1)

2 In linear programming problems the method is somewhat stronger because of the interaction between primal and dual problems. However no objective function is specified in (1.1). See also Telgen [6].
This is a system of linear inequalities. Suppose that all redundant inequalities can be removed in polynomially bounded time. If the original LP-problem has a finite solution, then all remaining inequalities are binding and pass through one point, so we can solve the remaining system by one matrix inversion of an \((m + n) \times (m + n)\) matrix. Note that it is not necessary to remove all redundant constraints from (6.2). It is sufficient to remove all strict redundant constraints from (6.2), since weak redundant constraints also pass through the only feasible solution. The resulting system of \(q \geq m + n\) inequalities in \(m + n\) variables can be solved by computing one generalized inverse for a \(q \times (m + n)\) matrix.

It follows that the general linear programming problem and the problem of identifying strict redundant constraints are polynomially related and thereby equally difficult in the sense of computational complexity theory (Aho, Hopcroft and Ullman [1]).

It remains an open question if a polynomial-time algorithm for the linear programming problem exists or not. Certainly, the simplex algorithm does not qualify as such; Klee and Minty [4] have shown that this method requires exponential time on certain weird polytopes under a variety of pivot selection rules. On the other hand, there is some theoretical evidence that unlike many other combinatorial optimization problems linear programming is not NP-hard (Aho, Hopcroft and Ullman [1]). Thus, the existence of a polynomial-time algorithm for its solution would not imply similarly efficient algorithms for a host of problems that are notorious for their computational intractability such as general 0-1 programming, the travelling salesman problem, etc. The complexity status of linear programming remains one of the major open problems in computational complexity theory.
ACKNOWLEDGEMENT

The author is greatly indebted to professors Gal (RWTH Aachen) for the many stimulating discussions, especially on the material contained in section 4, and Rinnooy Kan (EUR) for valuable comments on section 6.
REFERENCES


Reports 1977

7700  List of Reprints, nos. 179-194; List of Reports, 1976.
7701/M  "Triangular - Square - Pentagonal Numbers", by R.J. Stroeker.
7702/ES  "The Exact MSE-Efficiency of the General Ridge Estimator relative to OLS", by R. Teekens and P.M.C. de Boer.
7703/ES  "A Note on the Estimation of the Parameters of a Multiplicative Allocation Model", by R. Teekens and R. Jansen.
7707/S  "A Note on Theil's Device for choosing a Blus Base", by C. Dubbelman.
7710  "List of Reprints, nos. 195-199; Abstracts of Reports First Half 1977".
7711/M  "Degenerating Families of Linear Dynamical System I", by M. Hazewinkel.
7712/M  "Twisted Lubin-Tate Formal Group Laws, Ramified Witt Vectors and (Ramified) Artin-Hasse Exponential Mappings", by M. Hazewinkel.
7715/M  "Application of Non-Linear Programming to Plane Geometry", by R. Stroeker.
7716/S  "Stochastic Compactness of Sample Extremes" by L. de Haan and G. Ridder.