COMPUTER ASSISTED MODELLING OF
LINEAR, INTEGER AND SEPARABLE PROGRAMMING PROBLEMS
by
K. Darby-Dowman,
C. Lucas and
G. Mitra
$\begin{array}{ll}\text { Presented to: } & \text { NATO Advanced Study Institute on Computational } \\ \text { Mathematical Programming - July 1984-at } \\ & \text { Bad Windsheim, West Germany. }\end{array}$

w9259490

## ABSTRACT

For mathematical programming (MP) to have greater impact upon the decision making process, MP software systems must offer suitable support in terms of model communication and modelling techniques. In this paper modelling techniques that allow logical restrictions to be modelled in integer programming terms are described and their implications discussed. In addition it is demonstrated that many classes of non-linearities which are not variable separable may be reformulated in piecewise linear form. It is shown that analysis of bounds is necessary in the following three important contexts: model reduction, formulation of logical restrictions as $0-1$ mixed integer programs and reformulation of nonlinear programs as variable separable programs, It is observed that as well as incorporating an interface between the modeller and the optimiser there is a need to make available to the modeller software facilities which support the modelling techniques described here.

# COMPUTER ASSISTED MODELLING OF LINEAR, INTEGER AND SEPARABLE PROGRAMMING PROBLEMS. 

1. Introduction and Background.
2. Statement of the LP Model.
3. Analysis of Bounds for Linear Forms.
4. Representation of Logical Restrictions and Related Techniques
5. Strategies for Separating Variables in Nonlinear Programming Problems.
6. Discussion

References.

## COMPUTER ASSISTED MODELLING OF LINEAR, INTEGER AND SEPARABLE PROGRAMMING PROBLEMS

## 1. Introduction and Background

Modelling of mathematical programs and their computational solution are two salient activit ies in the exploitation of mathematical programming as a decision tool. Over the last thirty years or so substantial efforts have been devoted to the development of efficient algorithms for large scale applications. Efficient and robust computational algorithms are now well documented in the literature [1]. Most major computer manufacturers such as IBM (MPSX)[2], CDC (APEX) [3], UNIVAC (FMPS) [4] or software houses specializing in this area such as SCICON (SCICONIC) [5], KETRON (MPSIII) [6], have developed mathematical programming systems for the solution of linear and integer programming problems. Despite the availability of such software the use of mathematical programming as a decision making tool has not had the impact expected by dedicated practitioners. One reason for this state of affairs is that the availability and scope of good modelling support software for mathematical programming has not kept pace with developments both in computational software and computer technology in general. Various modelling systems such as MGRW [7], MAGEN [8], GAMMA III [9], MGG/RWG [10] DATAFORM [11], UIMP [12], LOGS[13], have been developed. These systems and others are primarily designed to ease the task of communicating a mathematical programming model to a computer, of documenting the model, and of creating solution reports. To date, however, most of these systems are based on a procedural language. Many of the current generation of applications systems are designed to be used by the problem owners themselves rather than by specialist intermediaries. Problem owners are consequently becoming more sophisticated in their use of computer supported mode 1ling. For these users, the requirements of a procedural language present an unnecessary hurdle. Modern interactive computing methods on the other hand, present the opportunity to design integrated easy to use systems.

The type of modelling support discussed so far assumes that a model has already been constructed. There is also great scope for support software to assist in formulating (and if necessary automatically reformulating) models in such a way that they can be solved by standard mathematical programming codes.

An experimental computer assisted mathematical programming system (CAMPS) is under development by the authors and Mr. M. Tamiz [14]. Within this integrated system it is possible to construe t, solve and analyse linear and some classes of non-Iinear problems. The des ign object ives of CAMPS and an outline description of its use are discussed in [15]

In this paper several issues relating to the formulation of non-linear ities in a way that can be handled by standard mathematical programming systems are discussed.

The contents of this paper are organized as follows. In section 2 the steps involved in formulating the underlying LP model are introduced and the notation defined. Analysis of bounds for linear forms is well known in the context of model reduction [15], [16 ]. The bound analysis results pertinent to integer and separable programming are presented in section 3. Some of the principles and methods underlying the formulation of the logical constraints using zero-one variables are outlined in section 4. Strategies for separating variables to represent a wide range of nonlinear programming problems are presented and discussed in section 5. Finally it is concluded in section 6 that the techniques discussed in this paper allow a modeller considerable scope in applying mathematical programming in practice. The materials contained in section $3,4,5$ are not new, however, in our analysis of the computer support for modelling we present a different focus on the underlying modelling principles and structure of these problems.

## 2. Statement of the LP Model

In order to derive a mathematical statement of the model one has to formally define the matrix elements of the constraint relations. In order to do this it is necessary to define the subscripts and their ranges. The matrix elements themselves may be derivedout of tabular input information relating to the problem. The sequence of steps leading to the derivation of a model naturally emerges and is set out below.

Step 1 Define the subscripts and their ranges (sets and dimensions).

Step 2 Define model variab les, constraints and the matrix coefficients in terms of the subscripts defined in step 1.

Step 3 Specify the linear relationships in a row wise fashion which connect the $i$ terns defined in step 2.

In its most general form an LP model can be stated in the following way:

- Subscripts, Ranges:

$$
\mathrm{i}=1, \ldots, \mathrm{~m}, \mathrm{j}=1, \ldots, \mathrm{n}
$$

- Variables, constraints, coefficients:

$$
\begin{aligned}
& x: x_{j}, j=1 \ldots n, r: r_{i}, i=1 \ldots m, d: d_{j}, j=1, \ldots n, \\
& c: c_{j} j=1 \ldots n, b: b_{i}, i=1 \ldots . m, \\
& A: a_{i j}:, i=1 \ldots m, j=1 \ldots n .
\end{aligned}
$$

- Linear objective function and constraints:

$$
\operatorname{Max} \sum_{j=1}^{n} c_{j} \mathrm{x}_{\mathrm{j}}
$$

subject to $\quad r_{i}=\sum_{j=1}^{n} a_{i j}{ }^{x}{ }_{j} \rho_{i} b_{i} \quad, \quad i=1, \ldots \ldots ., m$
where $\rho_{i}$ is an (in)equality relation of the form " $\leq "$, " $\geq$ " or " $=$ "
and . $\mathrm{d}_{\mathrm{j}}: \ell_{\mathrm{j}} \leq \mathrm{x}_{\mathrm{j}} \leq \mathrm{u}_{\mathrm{j}} \quad, \quad \mathrm{j}=1, \ldots \ldots, \mathrm{n}$.
where $\quad \ell_{j}$ may br $-\infty$ or finite $u_{j}$ may be $+\infty$ or finite.
3. Analysis of Bounds for Linear Forms
3.1 Use of Analysis in Model Reduction

Consider the restrictions $\mathrm{r}_{\mathrm{i}}$ and $\mathrm{d}_{\mathrm{i}}$ of the linear programming problem set out in (1) and discussed in section 2. Express these as two sets R and D of Linear Form constraints and Structural constraints respectively.

$$
\begin{align*}
& R=\left\{\left(x_{1}, \ldots \ldots x_{n}\right) \mid \sum_{j=1}^{n} \quad a_{i j} x_{j} \rho_{i} b_{i}, i=1, \ldots \ldots . . ., m\right\}  \tag{2}\\
& D=\left\{\left(x_{1}, \ldots x_{n}\right) \mid \quad \ell_{i} \leq x_{i} \leq u_{i}, \quad j=1, \ldots, n\right\} \tag{3}
\end{align*}
$$

It is well known [16],[17], that by considering the constraints sets R and D logically and iteratively, in many real life problems one may deduce the following:
(i) whether a constraint in set R is redundaut,
(ii) whether a constraint from set $R$ may be removed and replaced by a tighter bound in the set $D$,
(iii) whether a bound in the set D is redundant.

All these results follow from the analysis of the bounds on the linear forms .
3.2 An Analysis of the Linear Form

Let

$$
\begin{equation*}
F_{i}=\sum_{j=1}^{n} a_{i j} x_{j}, i=1, \ldots \ldots \ldots . ., m \tag{4}
\end{equation*}
$$

denote the ith linear form.
Introduce two index sets $\mathrm{P}_{\mathrm{i}}$, and $\mathrm{N}_{\mathrm{i}}$, (column indices of the positive and negative coefficients of the row i) such that

$$
\begin{equation*}
\mathrm{P}_{\mathrm{i}}=\left\{\mathrm{j} \mid \mathrm{a}_{\mathrm{i} \mathrm{j}}>0\right\}, \mathrm{N}_{\mathrm{i}}=\left\{\mathrm{j} \mid \mathrm{a}_{\mathrm{i} \mathrm{j}}<0\right\}, \mathrm{i}=1, \ldots \ldots \ldots, \mathrm{~m} \tag{5}
\end{equation*}
$$

Let $\quad \mathrm{L}_{\mathrm{i}} \leq \mathrm{F}_{\mathrm{i}} \leq \mathrm{U}_{\mathrm{i}} \quad, \mathrm{i}=1, \ldots \ldots \ldots, \mathrm{~m}$
denote the bounds on the linear form $F_{i}$; then from the definition of the structural bounds $\left(\ell_{\mathrm{j}} \leq \mathrm{x}_{\mathrm{j}} \leq \mathrm{u}_{\mathrm{j}}\right)$ the following is easily deduced:

$$
\begin{align*}
& U_{i}=\sum_{j \in p_{i}} a_{i j} u_{j}+\sum_{j \in N_{i}} a_{i j}{ }_{j},  \tag{7}\\
& L_{i}=\sum_{j \in p_{i}} a_{i j}{ }^{\ell}{ }_{j}+\sum_{j \in N_{i}} a_{i j}{ }^{u}, \tag{8}
\end{align*}
$$

In any of the following cases, the ith Linear Form constraint is redundant and may be removed from the problem
(a) $\quad \rho_{i}$ is " $\leq "$ and $U_{i} \leq b_{i}$,
(b) $\quad \rho_{i}$ is " $\geq$ " and $L_{i} \geq b_{i}$,

For a full discussion of these aspects of reduction the reader should refer to [16].

### 3.3 Examples

Example 1 A Redundant Constraint
Let the constraint sets R and D be as defined below.

$$
\begin{align*}
& \mathrm{R}=\left\{\left(\mathrm{x}_{1}, \mathrm{x}_{2}, \mathrm{x}_{3}\right) \mid \mathrm{x}_{1}+2 \mathrm{x}_{2}-\mathrm{x}_{3} \leq 11\right\}  \tag{9}\\
& \mathrm{D}-\left\{\left(\mathrm{x}_{1}, \mathrm{x}_{2}, \mathrm{x}_{3}\right) \mid 0 \leq \mathrm{x}_{1} \leq 1,0 \leq \mathrm{x}_{2} \leq 2,0 \leq \mathrm{x}_{3} \leq 4\right\}
\end{align*}
$$

The bounds on the linear form $\mathrm{F}_{\mathrm{i}}$ may be deduced as

$$
\mathrm{L}_{1}=-4, \quad \mathrm{U}_{1}=5
$$

We have $U_{1}<b_{1}$, hence the constraint is redundant.
Example 2 Tightening of a Bound
Let the constraints sets R and D be as defined below

$$
\begin{align*}
\mathrm{R} & =\left\{\left(\mathrm{x}_{1}, \mathrm{x}_{2}, \mathrm{x}_{3}\right) \mid \mathrm{x}_{1}+\mathrm{x}_{2}-2 \mathrm{x}_{3}=2\right\}  \tag{10}\\
\mathrm{D} & =\left\{\left(\mathrm{x}_{1}, \mathrm{x}_{2}>\mathrm{x}_{3}\right) \mid 0 \leq \mathrm{x}_{1} \leq 1,0 \leq \mathrm{x}_{2} \leq 3,0 \leq \mathrm{x}_{3} \leq 4\right\}
\end{align*}
$$

Since $a_{13}<0$ and $P_{1}$ is " $="$ an improved bound on $x_{3}$ is given by

$$
\mathrm{x}_{3} \leq \frac{\left(\mathrm{b}_{1}-\mathrm{U}_{1}\right)}{\mathrm{a}_{13}}
$$

$\mathrm{U}_{1}=4, \mathrm{~b}_{1}=2, \mathrm{a}_{13}=-2$, hence $\mathrm{x}_{3} \leq 1$ is the new bound which may be now introduced in the set $D$.

### 3.4 General Observations

It is pertinent at this stage to make the following observations concerning the bound analysis and its application in other contexts.
(i) $\quad L_{i}$, may be $-\infty$ or finite and $U_{i}$ may be $+\infty$ or finite. However, for finite values of $\ell_{i}, u_{i}, \quad j=1, \ldots . n, \quad i t$ follows from (7), (8) that $L_{i}, U_{i}$ are finite.
(ii) If the Linear Form constraints are connected by logical restrictions then $L_{i}, U_{i}$ values as necessary may be employed to (re)formulate these as $0-1$ mixed integer programs.
(iii) The derived bounds may be used in the improved reformulation and partial solution of integer programs.
(iv) It is not well known and rarely discussed in the literature that this analysis constitutes an essential part of any procedure for the reformulation of nonlinear, not variable separable functions into variable separable functions with arguments defined between upper and lower bounds.

The consequences of these observations in relation to integer and separable modelling are discussed in the following sections.
4. Representation of Logical Restrictions and Related Techniques.

### 4.1 Preliminary Considerations and Notation

It is well known that a large range of logical relationships connecting variables and constraint sets may be represented as integer or mixed integer programs. The authors have not come across any one source text where the underlying principles have been presented in a unified framework. However, most of the basic principles may be found in [18],[3], [19].

Let
$\begin{aligned} \Delta_{\mathrm{i}} \mathrm{i}= & 1,2, \ldots \text { denote logical variables which may take } \\ & \text { values. TRUE, or .FALSE., and }\end{aligned}$
$\delta_{\mathrm{i}} \mathrm{i}=1,2, \ldots \quad$ denote $0-1$ integer variables.
Define the following conventions and symbols for logical operators.
$\delta_{\text {I }}$ takes the value 1 , if and only if $\Delta_{i}$ is .TRUE., and 0 , if and only if $\Delta_{i}$ is -FALSE.

V denotes inclusive .OR.
$\dot{\mathrm{V}}$ denotes exclusive .OR.
\& denotes .AND.
$\equiv$ denotes equivalence...' if and only if '
Representing.OR.

If the condition $\Delta_{1} \mathrm{~V} \Delta_{2} \mathrm{~V} \Delta_{3} \mathrm{~V} \Delta_{4}$ is required to hold then this can be represented by the constraints

$$
\begin{equation*}
\delta_{1}+\delta_{2}+\delta_{3}+\delta_{4} \geq 1 \tag{11}
\end{equation*}
$$

Similarly exclusive .OR. relations as in the requirement $\Delta_{1} \dot{V} \Delta_{2} \dot{V} \Delta_{3} \dot{\mathrm{~V}} \Delta_{4}$ can be represented by the constraint

$$
\begin{equation*}
\delta_{1}+\delta_{2}+\delta_{3}+\delta_{4}=1 \tag{12}
\end{equation*}
$$

Let $Y$ denote a logical variable and $y$ the corresponding $0-1$ variable and let these be related in the same way as $\Delta_{i}$ and $\delta_{i}$ are related to each other.

Then the condition $: \mathrm{Y}$ is .TRUE, if and only if $\Delta_{1} \mathrm{~V} \Delta_{2} \mathrm{~V} \Delta_{3} \ldots \Delta_{\mathrm{k}}$ is .TRUE. (which is expressed as $\mathrm{Y}=\Delta_{1} \mathrm{~V} \Delta_{2} \mathrm{~V} \ldots . \Delta_{\mathrm{k}}$ ), can be represented by the constraint

$$
\begin{equation*}
-(\mathrm{k}-1) \leq \delta_{1}+\delta_{2}+\ldots \delta_{\mathrm{k}}-\mathrm{ky} \leq 0 \tag{13}
\end{equation*}
$$

Representing .AND.
The logical condition

$$
\mathrm{Y}=\left(\begin{array}{lllllll}
\Delta_{1} & \& & \Delta_{2} & \& & \ldots & \& & \Delta_{\mathrm{k}}
\end{array}\right)
$$

can be represented by the constraint

$$
0 \leq \delta_{1}+\delta_{2}+\ldots+\delta_{\mathrm{k}}-\mathrm{ky} \leq \mathrm{k}-1
$$

For other logical relationships and a discussion of the general form the readers should refer to \{3].
p-fold Alternatives
The general forms of the relations (11 ), (12) may be stated as

$$
\begin{equation*}
\delta_{1}+\delta_{2}+\ldots+\delta_{k} \geq 1 \tag{14}
\end{equation*}
$$

and

$$
\begin{equation*}
\delta_{1}+\delta_{2}+\ldots+\delta_{k}=1 \tag{15}
\end{equation*}
$$

which represent the inclusive and exclusive .OR. respective ly of k : logical variables.

Now consider the relations

$$
\begin{equation*}
\delta_{1}+\delta_{2}+\ldots+\delta_{k} \geq P \tag{16}
\end{equation*}
$$

and

$$
\begin{equation*}
\delta_{1}+\delta_{2}+\ldots+\delta_{k}=p \tag{17}
\end{equation*}
$$

where p is an integer and $1 \leq \mathrm{p} \leq \mathrm{k}$. The relation (16) represents the statement "p or more alternatives hold at any time" and (17) represents the statement "exactly $p$ alternatives hold at any time".
4.2 Logically Relating the Linear Form Constraints

A linear form constraint involving $n$ variables represents a point set in $E^{n}$. If a number of these are stated and need to be satisfied then these invoke the logical .AND. operation.

Thus for

$$
\left.\begin{array}{l}
\mathrm{R}_{1}=\left\{\left(\mathrm{x}_{1} \ldots \mathrm{x}_{\mathrm{n}}\right) \mid\right. \\
\vdots  \tag{18}\\
\vdots \\
\vdots \\
\mathrm{n}=1 \\
\left.\mathrm{a}_{1 \mathrm{j}} \mathrm{x}_{\mathrm{j}} \leq \mathrm{b}_{1}\right\} \\
\mathrm{R}_{\mathrm{m}}=\left\{\left(\mathrm{x}_{1} \ldots \mathrm{x}_{\mathrm{n}}\right) \mid\right.
\end{array} \sum_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{a}_{\mathrm{m} \mathrm{j}} \mathrm{x}_{\mathrm{j}} \leq \mathrm{b}_{\mathrm{m}}\right\}, ~ l
$$

the constraint set

$$
\begin{equation*}
\mathrm{R}=\mathrm{R}_{1} \quad \& \quad \mathrm{R}_{2} \quad \& \quad \ldots \quad \mathrm{R}_{\mathrm{m}} \tag{19}
\end{equation*}
$$

is stated as

$$
\begin{equation*}
R=\left\{\left(x_{1} \ldots \ldots x_{n}\right) \mid \sum_{j=1}^{n} a_{i j} x_{j} \leq b_{i}, i=1, \ldots \ldots ., \quad m\right\} \tag{20}
\end{equation*}
$$

It may be observed that $R$, the intersection of $R_{1}, R_{2}, \ldots R_{m}$, is convex as $\mathrm{R}_{\mathrm{i}}, \quad \quad \mathrm{i}=1, \ldots, \mathrm{~m}$ are convex.
However, to represent the logical.OR. relation of these constraint sets $R_{1}, R_{2}, \ldots . R_{m}$ it is necessary to consider the structural constraint set

$$
\begin{equation*}
\mathrm{D}=\left\{\left(\mathrm{x}_{1} \ldots \mathrm{x}_{\mathrm{n}}\right) \mid \ell_{\mathrm{j}} \leq \mathrm{x}_{\mathrm{j}} \leq \mathrm{u}_{\mathrm{j}}, \mathrm{j}=1, \ldots, \mathrm{n}\right\} \tag{21}
\end{equation*}
$$

where some or all $\ell_{j}, u_{j} j=1, \ldots n$ are finite such that the bounds $U_{i}, i=1 \ldots m$ are finite. Also from the redundancy consideration it is required that $b_{i}<U_{I} \quad, \quad i=1, . ., m$.

To represent the inclusive .OR. relation

$$
\begin{equation*}
\mathrm{R}_{1} \quad \mathrm{~V}_{2} \quad \mathrm{R}_{2} \quad \mathrm{~V} \ldots \mathrm{R}_{\mathrm{m}} \tag{22}
\end{equation*}
$$

introduce the relations

$$
\begin{equation*}
\sum_{j=1}^{n} a_{i j} x_{j}-B_{i}\left(1-\delta_{i}\right) \leq b_{i}, i=1, \ldots \ldots \ldots \quad m \tag{23}
\end{equation*}
$$

and $\quad \sum_{i=1}^{m} \delta_{i} \geq 1$.
where $B$. is a finite value such that for $\delta_{i}=0, B_{i}+b_{i}$ is greater than or equal to the upper bound of
$F_{i}=\sum_{j=1}^{n} a_{i j} x_{j}$.
Thus any finite value;' for $B_{i}$ such that

$$
\begin{equation*}
\mathrm{B}_{\mathrm{i}}+\mathrm{b}_{\mathrm{i}} \geq \mathrm{U}_{\mathrm{i}} \quad, \quad \mathrm{i}=1, \ldots, \mathrm{~m} \tag{25}
\end{equation*}
$$

leads to a valid formulation. The exclusive .OR. and the two forms of p-fold alternatives are similarly obtained with (24) replaced by (26), (27), or (28) respectively

$$
\begin{align*}
& \sum_{\substack{i=1 \\
\mathrm{~m}}}^{\mathrm{m}} \delta_{\mathrm{i}}=1  \tag{26}\\
& \sum_{\substack{i=1}} \delta_{\mathrm{i}} \geq \mathrm{p}  \tag{27}\\
& \sum_{\mathrm{i}=1}^{\mathrm{m}} \delta_{\mathrm{i}}=\mathrm{p} \tag{28}
\end{align*}
$$

To illustrate these points, consider the following example taken from [18] and modified.

$$
\text { Let } \begin{align*}
& \mathrm{R}_{1}=\left\{\left(\mathrm{x}_{1}, \mathrm{x}_{2}\right) \mid\right. \\
& \mathrm{R}_{2}=\left\{\left(\mathrm{x}_{1}+\mathrm{x}_{2}, \mathrm{x}_{2}\right) \mid-\mathrm{x}_{1}+\mathrm{x}_{2} \leq 0\right\}  \tag{29}\\
& \mathrm{R}_{3}=\left(\left(\mathrm{x}_{1}, \mathrm{x}_{2}\right) \mid\right. \\
&\left.3 \mathrm{x}_{1}-\mathrm{x}_{2} \leq 8\right\}
\end{align*}
$$

and let $\mathrm{D}=\left\{\left(\mathrm{x}_{1}, \mathrm{x}_{2}\right) \mid 0 \leq \mathrm{x}_{1} \leq 5,0 \leq \mathrm{x}_{2} \leq 5\right\}$

Then
$S=R \& D=R_{1} \& R_{2} \& R_{3} \& D$ is as shown in Diagram 1.


The three bounds on the linear forms may be computed as

$$
U_{1}=10 \quad, \quad U_{2}=5, \quad U_{3}=15
$$

10. 

A formulation which uses the logical .OR. as well as .AND. relations such as $T=R_{1} V\left(R_{2} \& R_{3}\right)$ may be stated as

$$
\begin{gather*}
x_{1}+x_{2}-6\left(1-\delta_{1}\right) \leq 4, \\
-x_{1}+x_{2}-5\left(1-\delta_{2}\right) \leq 0, \\
3 x_{1}-x_{2}-7(1-\delta 2) \leq 8,  \tag{30}\\
\delta_{1}+\delta_{2} \geq 1
\end{gather*}
$$

$$
\text { and } \quad \delta_{1}, \delta_{2}=0,1
$$

The constraint region T in this case is as shown in Diagram 2.


Because of the inclusive. OR. relation the constraint region $T$ which is logically stated and represented by the mixed integer formuation (30) is not a convex region.
5. Strategies for Separating Variables in Non Linear Programming Problems
5.1 Linearization of Variable Separable Programming Problems.

The problem
$\operatorname{Max} \sum_{j=1}^{n} f_{j}\left(x_{j}\right)$
subject to $\sum_{j=1}^{n} g_{i j}\left(x_{j}\right) \leq b_{i}, \quad i=1, \ldots \ldots \ldots . . . . m$,
is a general statement of the variable separable programming problem. In order to carry out piecewise linear approximations to the objective and the constraint functions it is necessary to make two further assumptions concerning this problem.
(i) The functions $\mathrm{f}_{\mathrm{j}}\left(\mathrm{x}_{\mathrm{j}}\right), \mathrm{j}=1, . ., \mathrm{n}$
are all s ingle valued.
(ii) The arguments $\mathrm{x}_{\mathrm{j}}, \mathrm{j}=1, \ldots, \mathrm{n}$ of these functions have finite ranges $\left(\ell_{\mathrm{j}} \leq . \mathrm{x}_{\mathrm{j}} \leq \mathrm{u}_{\mathrm{j}} \quad, \quad \mathrm{j}=1, \ldots, \mathrm{n}\right)-$

The construction of piecewise linear approximat ions using weighting variables, convexity row, reference row and function row is well discussed in [18], [20], [22], [23].

### 5.2 An Analysis of Nonlinear Programming Test Problems

It has been claimed by proponents of the separable programming method of solving nonlinear programming problems that a large class of nonlinear (not variable separable) programming problems can be transformed into variable separable programming problems. In order to investigate the reality of this claim the comprehensive collection of nonlinear programming test problems which have been put together in [24] have been analysed.

Consider the test problems in the format

$$
\text { Maximise } \quad f\left(x_{1}, \ldots, x_{n}\right)
$$

subject to

$$
\begin{array}{ll}
g_{i}\left(x_{1} \ldots, x_{n}\right) \leq b_{i}, & i=1, \ldots, m \\
g_{i}\left(x_{1} \ldots, x_{n}\right)=b_{i}, & i=m_{i}+1, \ldots, m \tag{33}
\end{array}
$$

and

$$
\ell_{\mathrm{j}} \leq \mathrm{x}_{\mathrm{j}} \leq \mathrm{u}_{\mathrm{j}} \quad, \quad \mathrm{j}=1, \ldots, \mathrm{n} .
$$

The following types of objective functions $f(x)$ and constraint functions $\mathrm{g}_{\mathrm{i}}(\mathrm{x})$ are found in the set of problems.

## 12.

Objective function types
(i) Constant objective function ... function code C.
(ii) Linear objective function ... function code L.
(iii) Quadratic objective function ... function code Q .
(iv) Sum of squares objective function ... function code S.
(v) Generalized polynomial objective function ... function code P. This is of the form

$$
\begin{equation*}
f(x)=a_{0}+\sum_{i=1}^{n} a_{i} x_{i}+\sum_{i, j=1}^{n} a_{i j} x_{i} x_{j}+\sum_{i, j, k=1}^{n} a_{i j j k} x_{i} x_{j} x_{k}+\ldots \ldots \ldots \tag{34}
\end{equation*}
$$

It may be observed that in geometric programming problems [25] a more general form is introduced which is called the signomial function and is expressed as

$$
\begin{equation*}
\mathrm{f}(\mathrm{x})=\sum_{\mathrm{j} \in \mathrm{~J}} \mathrm{c}_{\mathrm{j}} \pi_{\mathrm{i}} x_{i}^{\mathrm{dij}} \tag{35}
\end{equation*}
$$

where $J$ is used to label the terms appearing in the signomial function. In (34) $a_{0}, a_{i}, a_{i j}$ etc, and in (35) $c_{j}, d_{i j}$ are given real values.
(vi) General function ... function code G.

## Constraint types

(i) No constraint ... code $U$
(ii) Only upper and lower bounds on the variables ... code B
(iii) Linear constraint functions ... code L
(iv) Quadratic constraint functions ... code Q
(v) Generalized Polynomial constraint functions ... code P This is of the same form as (34) or (35).
(vi) Generalized constraint functions ... code G.

The frequency distribution of the 115 test problems is set out in Table 1 In [24] the problems are numbered from 1 to 119 , however, there are no problems numbered $58,82,94,115$ !

Objective Function Codes


Table 1

### 5.3 Manipulation of Non-Linear Functions to Variable Separable Form.

The principal motivation of deriving variable separable formulations of non-linear functions is that such formulations may be approximated using piecewise linear forms. Consequently a standard mathematical programming system (e.g. MPSX) can be used to solve these classes of non-linear programming problems. In order to apply a piecewise linear approximation it is required that the variables of the separable formulation, which are derived from the original non-linear functions, be bounded. It is therefore necessary to apply a bound analysis to determine these bounds. In practical applications it is possible to impose realistic bounds on any unconstrained variables which may appear in the problem.

In this section it is illustrated, by means of examples, that a wide range of non-linear functions may be expressed in a variable separable form. In addition the corresponding bound analyses (essential for piecewise linear approximations) are presented. It may be observed that problems in which the objective function has code $\mathrm{C}, \mathrm{L}$ or S and constraints with codes U , B or L are clearly in variable separable form.

## Product Term

A product term, $x_{1} x_{2}$, may be replaced by $\left(y_{1}^{2}-y_{2}^{2}\right)$ with the additional constraints $\mathrm{y}_{1}=1 / 2\left(\mathrm{x}_{1}+\mathrm{x}_{2}\right)$ and $y_{2}=1 / 2\left(\mathrm{x}_{1}-\mathrm{x}_{2}\right)$. If $\left(\ell_{\mathrm{i}} \leq \mathrm{x}=\leq \mathrm{u}_{\mathrm{i}}\right)$ then, given finite $\ell_{i}$ and $u_{i}$, finite bounds $L_{i}$ and $U_{i}$ may easily be derived such that $\left(L_{i} \leq y_{i} \leq U_{i}\right), \quad i=1,2$.
By repeated application of this technique a variable separable formulation of a higher order product term may be obtained.

## Quadratic Function $\phi$

For a general quadratic function, $\phi\left(\mathrm{x}_{1}, \ldots \mathrm{x}_{\mathrm{n}}\right)$ a more compact variable separable formulation may be obtained.
Let $\phi\left(x_{1}, \ldots \ldots . . x_{n}\right)=\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} q_{i j} x_{i} x_{j}$
$\operatorname{Replace} \phi\left(\mathrm{x}_{1}, \ldots \ldots . . \mathrm{x}_{\mathrm{n}}\right)$ by $\psi\left(\mathrm{y}_{1}, \ldots \ldots . \mathrm{y}_{\mathrm{r}}\right)=\sum_{\mathrm{k}=1}^{\mathrm{r}} \mathrm{d}_{\mathrm{k}} \mathrm{y}_{\mathrm{k}}^{2}$
With the constraints

$$
\begin{equation*}
\mathrm{y}_{\mathrm{k}}=\sum_{\mathrm{j}=1}^{\mathrm{n}} \mathrm{q}_{\mathrm{k}}^{\prime} \mathrm{x}_{\mathrm{j}} \quad \mathrm{k}=1, \ldots \ldots . . \mathrm{r} \tag{37}
\end{equation*}
$$

where $r$ is the rank of the symmetric matrix $Q=\left\{q_{i j}\right\}$.
The coefficients $\mathrm{q}_{\mathrm{kj}}{ }^{\prime}$. and $\mathrm{d}_{\mathrm{k}}$ can be determined by applying a standard method such as Lagrange's Reduction.

Given finite bounds $\ell_{\mathrm{j}}$ and $\mathrm{u}_{\mathrm{j}}$ on $\mathrm{x}_{\mathrm{i}}, \mathrm{j}=1, \ldots \mathrm{n}$, finite bounds $L_{k}$ and $U_{k}$ on $y_{k}, k=1, \ldots r$, may be simply derived by considering the linear forms (37), thus enabling a piecewise linear approximation to be used.

Ratio of Linear Forms
Let $H^{\prime}=\sum_{j=1}^{n} h^{\prime}{ }_{j}{ }^{x}{ }_{j}$ and $\quad H^{\prime}{ }^{\prime}=\sum_{j=1}^{n} h^{\prime \prime}{ }_{j}{ }^{x}{ }_{j}$.

The expression ( $\mathrm{H}^{\prime} / \mathrm{H}^{\prime \prime}$ ) may be manipulated in the following way. Replaced $\left(H^{\prime} / H^{\prime \prime}\right)$ by $y_{1}$ and introduce the constraint $\sum_{j=1}^{n} h^{\prime}{ }_{j} x_{j}=\sum_{j=1}^{n} h^{\prime} x^{\prime}{ }_{j} y_{1}$. As discussed earlier a variable separable formulation may be obtained for the product terms of the constraint. The finite bounds on $x_{i}, j=1, \ldots n$, provide bounds on $\mathrm{H}^{\prime}$ and $\mathrm{H}^{\prime \prime}$ such that $\mathrm{L}^{\prime} \leq \mathrm{H}^{\prime} \leq \mathrm{U}^{\prime}$ and $\mathrm{L}^{\prime \prime} \leq \mathrm{H}^{\prime \prime} \leq \mathrm{U}^{\prime \prime}$ from which bounds on $y_{1}$ may be obtained. If $L^{\prime \prime}>0$ or $U^{\prime \prime}<0$, the bounds on $y_{1}$ are finite and a piecewise linear formulation can be applied.
Power Forms - Constant Base
Consider the term $\mathrm{ax}_{1}+\mathrm{x}^{2}$ where $\mathrm{a}>0$.
A variable separable formulation may be obtained by replacing $a^{x 1^{+}+x_{2}^{2}}$ by $y_{1}$ And introducing the constraint $\log \mathrm{y}_{1}=(\log \mathrm{a}) \cdot\left(\mathrm{x}_{1}+\mathrm{x}_{2}^{2}\right)$. The bounds $L_{1}$ and $U_{1}$ on can be derived from the bounds on $x_{1}$ and $x_{2}$.
Power Forms - Variavle Base
Consider the term $\mathrm{x}_{1}^{\mathrm{x}^{2}}$. This term can be handled using the substitution

$$
\begin{align*}
& \mathrm{y}_{1}=10^{\mathrm{yx} 2}  \tag{38}\\
& \mathrm{x}_{1}=10^{\mathrm{y} 2} \tag{39}
\end{align*}
$$

The constraint (38) can be handled using the techniques for product terms and constant base power forms discussed earlier. For constraint (39) it it necessary that $0<\ell_{1} \leq \mathrm{x}_{1} \leq \mathrm{u}_{1}$ from which the bounds on $\mathrm{y}_{2}$ are easily derived-
The range of functions illustrated above show that the only problems that cannot easily be formulated as variable separable lie in the class in which the objective or constraint code is G. However, most problems in this class can be transformed to a separable form without difficulty. To illustrate this point consider the following example.

### 5.4 An Example

Consider the problem [26].
Maximise $\mathrm{x}_{1}+2 \mathrm{x}_{2}+\mathrm{x}_{3}$
Subject

$$
\begin{gather*}
\mathrm{x}_{1} \mathrm{x}_{2}+\frac{\mathrm{x}_{2}}{1+\mathrm{x}_{1}} \mathrm{e}^{\mathrm{x}_{3}+\mathrm{x}_{3} \leq 20}  \tag{40}\\
\mathrm{x}_{1}+\mathrm{x}_{2}+\mathrm{x}_{3} \leq 4 \tag{41}
\end{gather*}
$$

and

$$
\begin{equation*}
\mathrm{x}_{1}, \mathrm{x}_{2}, \mathrm{x}_{3} \geq 0 \tag{42}
\end{equation*}
$$

From restriction (41), (42) it follows that

Rewrite

$$
\begin{align*}
& 0 \leq x_{1}, x_{2}, x_{3} \leq 4 \\
& \frac{x_{2}}{1+x_{1}}=x_{4} \text { or } x_{2}-x_{4}-x_{1} x_{4}=0 \tag{44}
\end{align*}
$$

Now from (43) and (44) $\quad \ell_{4} \leq x_{4} \leq u_{4}$ where

$$
\begin{equation*}
\ell_{4}=0 \quad, \quad u_{4}=4 \tag{45}
\end{equation*}
$$

The constraint (40) can be reexpressed as

$$
\begin{array}{ll} 
& \mathrm{x}_{1} \mathrm{x}_{2}+\mathrm{x}_{4} \mathrm{y}_{1}+\mathrm{x}_{3} \leq 20 \\
\text { and } & \mathrm{y}_{1}=\mathrm{e}^{\mathrm{x}} 3 \tag{47}
\end{array}
$$

From (43) and (47) the following bounds are derived

$$
\mathrm{e}^{\circ}=1 \leq \mathrm{y}_{1}<\mathrm{e}^{4}=54.598
$$

Thus the given problem may be restated as
Maximize $\mathrm{x}_{1}+2 \mathrm{x}_{2}+\mathrm{x}_{3}$
subject to $\quad \mathrm{x}_{1} \mathrm{x}_{2}+\mathrm{x}_{4} \mathrm{y}_{1}+\mathrm{x}_{3} \leq 20$
$\mathrm{x}_{2}-\mathrm{x}_{4}-\mathrm{x}_{1} \mathrm{x}_{4}=0$

$$
\begin{gather*}
\mathrm{y}_{1}-\mathrm{e}^{\mathrm{x} 3}=0  \tag{48}\\
\mathrm{x}_{1}+\mathrm{x}_{2}+\mathrm{x}_{4} \leq 4
\end{gather*}
$$

and

$$
0<\mathrm{x}_{1}, \mathrm{x}_{2}, \mathrm{x}_{3}, \mathrm{x}_{4} \leq 4, \quad 1.0 \leq \mathrm{y}_{1} \leq 54.598
$$

The product terms are thus re-expressed as

$$
\mathrm{x}_{1} \mathrm{x}_{2}=\mathrm{z}_{1}^{2}-\mathrm{z}_{2}^{2}, \mathrm{x}_{4} \mathrm{y}_{1}=\mathrm{z}_{3}^{2}-\mathrm{z}_{4}^{2}, \mathrm{x}_{1} \mathrm{x}_{4}=\mathrm{z}_{5}^{2}-\mathrm{z}_{6}^{2}
$$

which leads to the full separable programming formulation:
Maximise $\quad \mathrm{x}_{1}+2 \mathrm{x}_{2}+\mathrm{x}_{3}$
Subject to

$$
\begin{gathered}
z_{1}^{2}-z_{2}^{2}+z_{3}^{2}-z_{4}^{2}+x_{3} \leq 20 \\
x_{2}-x_{4}-z_{5}^{2}+z_{6}^{2}=0 \\
y_{1}-e^{x_{3}}=0 \\
x_{1}+x_{2}+x_{4} \leq 4 \\
{ }_{2}^{1} x_{1}+{ }_{2}^{1} x_{2}-z_{1}=0 \\
{ }_{2}^{1} x_{1}+{ }_{2}^{1} x_{2}-z_{2}=0 \\
{ }_{2}^{1} x_{4}+{ }_{2}^{1} y_{1}-z_{3}=0 \\
{ }_{2}^{1} x_{4}+{ }_{2}^{1} y_{1}-z_{4}=0 \\
{ }_{2}^{1} x_{1}+{ }_{2}^{1} x_{4}-z_{5}=0 \\
{ }_{2}^{1} x_{1}+{ }_{2}^{1} x_{4}-z_{6}=0 \\
0 \mathrm{z}_{\mathrm{z}} \mathrm{x}_{1} \cdot \mathrm{x}_{2} \cdot \mathrm{x}_{3} \cdot \mathrm{x}_{4} \leq 4, \quad 1 \leq \mathrm{y}_{1} \leq 54.598
\end{gathered}
$$

with $\ell_{\mathrm{i}}^{\mathrm{Z}} \leq \mathrm{z}_{\mathrm{i}} \leq \mathrm{u}_{\mathrm{i}}^{\mathrm{Z}}$ as the easily derived bounds on the $\mathrm{z}_{\mathrm{i}}, \mathrm{i}=1, \ldots \ldots \ldots . .6$

## 17.

## 6. Discussion

Most modelling systems focus on the task of creating model data and offer little support towards deriving the model itself. The analysis of bounds plays a key role in model reduction, for mulation of logical restrictions and reformulation of non-linear programs as variable separable programs. A mathematical programming modelling system should contain facilities to assist the user to specify his model and reduce the chore of algebraic manipulation. This aspect is particularly important for problem owners who are capable of describing their models precisely but who nevertheless may not be experienced in reformulation techniques and may not be skilled in algebraic manipulation. Computer support in these areas offers increased scope and applicability of mathematical programming.

## References

1. Balinski, M.L., and Hellerman, E., editors, Computational Practice in Mathematical Programming, Mathematical Programming Study 4, North Holland, 1975.
2. IBM Corporation, Mathematical Programming System Extended, MPSX/370, Reference Manual, SM19-1095-1, 1976.
3. Control Data Corporation, APEX III Reference Manual, Version 1.2, No. 76070000, Revisions G, 1979.
4. Sperry Univac, FMPS: Functional Mathematical Programming System, User Manual for UNIVAC 1108 Computers, 1978.
5. SCICON, Computer Services, SCICONIC User Manual, 1978.
6. Ketron Inc., MPS III User Manual, 1980.
7. IBM Corporation, MGRW Program Reference Manual, Program SH19-5014, 1977.
8. Haverly Systems, MAGEN: Reference Manual, 1977.
9. Sperry Univac, GAMMA3 User Manual for UNIVAC 1108 Computers, 1978.
10. SCICON Computer Services, MGG User Guide, RWG User Guide, 1975.
11. Ketron Inc., MPSIII DATAFORM : User Manual, 1980.
12. Mitra, G., and Ellison, E.F.D., User Interface to Mathematical Programming : UIMP, ACM Transactions on Mathematical Software, Vol. 8, No. 3, p229-255, 1982.
13. Brown, R.W., Northup, W.D., and Shapiro, J.F., LOGS: A Modelling and Optimization System for Business Planning, in Computer Assisted and Decision Making, Ed. G. Mitra, North Holland, 1984.
14. Mitra, G., and Tamiz, M., A Suite of FORTRAN based LP Subroutines : FORTLP, User Speci tication, Brunei University Computer Unit, 1982.
15. Lucas, C., Mitra, G., and Darby-Dowman, K., Modelling of Mathematical Programs: An Analysis of Strategy and an Outline Description of a Computer Assisted System, TR/09/33, Brunel University and submitted to Annals of O.R.
16. Brearley, A.L., Mitra, G., and Williams, H.P., Analysis of Mathematical Programming Models Prior to Applying the Simplex Algorithm, Mathematical Programming, Vol.8, pp 54-83, 1975.
17. Williams, H.P., A Reduction Procedure for Linear and Integer Programming Models, in Redundancy in Mathematical Programming, Edited by S. Zionts et al, pp 87-109, Springer Verlag, 1983.
18. Williams, H.P., Model Building in Mathematical Programming, John Wiley \& Sons, 1978.

19 Simonnard, M., Linear Programming, Prentice Hall Internat ional, 1966.
20. Mitra, G., Theory and Application of Mathematical Programming, Academic Press, 1976.
21. Bradley, S.P., Hax, A.C., and Magnanti, T.L. Applied Mathematical Programming, Addison-Wesley, 1977.
22. Beale, E.M.L., Mathematical Programming in Pract ice, Pitman 1968.
23. Hadley, G.H., Nonlinear Programming, Addison Wesley, 1964.
24. Hock, W., and Schittkowski, K. Test Examples for Nonlinear Programming Codes, Springer Verlag, 1981.
25. Dembo, R.S., A set of Geometric Programming Test Problems and their Solutions, Mathemat ical Programming, Vol. 10, pp 192-213, 1976.


