TR/08/84

August, 1984.
Updated: October, 1984.

COMPUTER ASSISTED MODELLING OF LINEAR, INTEGER AND SEPARABLE PROGRAMMING PROBLEMS

by

K. Darby—Dowman, C. Lucas and

G. Mitra

Presented to: NATO Advanced Study Institute on Computational Mathematical Programming - July 1984 - at Bad Windsheim, West Germany.

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.

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 activities 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 modelling. 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 construct, solve and analyse linear and some classes of non—linear problems. The design objectives of CAMPS and an outline description of its use are discussed in [15].

In this paper several issues relating to the formulation of non—linearities 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:

  $i = 1, \ldots, m,\; j = 1, \ldots, n.$

- Variables, constraints, coefficients:

  $x : x_j$, $j = 1 \ldots n$, $r : r_i$, $i = 1 \ldots m$, $d : d_j$, $j=1, \ldots n$,
  $c: c_j$ $j = 1 \ldots n$, $b : b_i$, $i = 1 \ldots m$,
  $A : a_{ij}$:, $i = 1 \ldots m, j = 1 \ldots n$.

- Linear objective function and constraints:

$$\text{Max} \sum_{j=1}^{n} c_j x_j \quad ,$$

subject to $r_i = \sum_{j=1}^{n} a_{ij} x_j \, \rho_i \, b_i$, $i = 1, \ldots\ldots, m$

where ρ_i is an (in)equality relation of the form "$\leq$" , "$\geq$" or "$=$" (1)

and . $d_j : \ell_j \leq x_j \leq u_j$, $j = 1, \ldots\ldots, n$.

where ℓ_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 r_i and d_j 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.

$$R = \{(x_1,......x_n) \mid \sum_{j=1}^{n} a_{ij}x_j \rho_i b_i, \ i=1,........,m\} \tag{2}$$

$$D = \{(x_1,...x_n) \mid \ell_j \le x_j \le u_j, \ j = 1,...,n\} \tag{3}$$

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

$$F_i = \sum_{j=1}^{n} a_{ij} x_j, \ i = 1,......... ., m \tag{4}$$

denote the ith linear form.

Introduce two index sets P_i, and N_i, (column indices of the positive and negative coefficients of the row i) such that

$$P_i = \{j \mid a_{ij} > 0\}, N_i = \{j \mid a_{ij} < 0\}, \ i = 1, ..., m \tag{5}$$

Let $L_i \le F_i \le U_i$, $i = 1,.........,m$ (6)
denote the bounds on the linear form F_i; then from the definition of the structural bounds ($\ell_j \le x_j \le u_j$)the following is easily deduced:

$$U_i = \sum_{j \in p_i} a_{ij} u_j + \sum_{j \in N_i} a_{ij} \ell_j , \tag{7}$$

$$L_i = \sum_{j \in p_i} a_{ij} \ell_j + \sum_{j \in N_i} a_{ij} u_j , \tag{8}$$

In any of the following cases, the ith Linear Form constraint is redundant and may be removed from the problem

(a) ρ_i is "$\leq$" and $U_i \leq b_i$,

(b) ρ_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.

$$R = \{(x_1,x_2,x_3) \mid x_1 + 2x_2 - x_3 \leq 11\}$$
$$D - \{(x_1,x_2,x_3) \mid 0 \leq x_1 \leq 1,\ 0 \leq x_2 \leq 2,\ 0 \leq x_3 \leq 4\} \qquad (9)$$

The bounds on the linear form F_i may be deduced as

$L_1 = -4$, $U_1 = 5$.

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

$$R = \{(x_1,x_2,x_3) \mid x_1 + x_2 - 2x_3 = 2\}$$
$$D = \{(x_1,x_{2>}x_3) \mid 0 \leq x_1 \leq 1,\ 0 \leq x_2 \leq 3,\ 0 \leq x_3 \leq 4\} \qquad (10)$$

Since $a_{13} < 0$ and P_1 is "=" an improved bound on x_3 is given by

$$x_3 \leq \frac{(b_1 - U_1)}{a_{13}}$$

$U_1 = 4$, $b_1 = 2$, $a_{13} = -2$, hence $x_3 \leq 1$ 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) L_i, may be $-\infty$ or finite and U_i may be $+\infty$ or finite. However, for finite values of ℓ_j, u_j, $j = 1,\ldots.n$, it 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

Δ_i $i = 1,2,\ldots$ denote logical variables which may take values . TRUE, or .FALSE., and

δ_i $i = 1,2,\ldots$ denote 0-1 integer variables.

Define the following conventions and symbols for logical operators.

δ_I takes the value 1, if and only if Δ_i is .TRUE.,

and 0, if and only if Δ_i is -FALSE.

V denotes inclusive .OR.

$\dot{V}$ denotes exclusive .OR.

& denotes .AND.

$\equiv$ denotes equivalence...' if and only if'

<u>Representing .OR.</u>

If the condition $\Delta_1 V \Delta_2 V \Delta_3 V \Delta_4$ is required to hold then this can be represented by the constraints

$$\delta_1 + \delta_2 + \delta_3 + \delta_4 \geq 1 \quad . \tag{11}$$

Similarly exclusive .OR. relations as in the requirement $\Delta_1 \dot{V} \Delta_2 \dot{V} \Delta_3 \dot{V} \Delta_4$ can be represented by the constraint

$$\delta_1 + \delta_2 + \delta_3 + \delta_4 = 1 \ . \tag{12}$$

Let Y denote a logical variable and y the corresponding 0-1 variable and let these be related in the same way as Δ_i and δ_i are related to each other.

Then the condition : Y is .TRUE, if and only if $\Delta_1 V \Delta_2 V \Delta_3 \ldots\ldots \Delta_k$ is .TRUE. (which is expressed as $Y = \Delta_1 V \Delta_2 V \ldots\ldots \Delta_k$), can be represented by the constraint

$$-(k - 1) \leq \delta_1 + \delta_2 + \ldots \delta_k - ky \leq 0 \tag{13}$$

Representing .AND.

The logical condition

$$Y = (\Delta_1 \ \& \ \Delta_2 \ \& \ \ldots \ \& \ \Delta_k)$$

can be represented by the constraint

$$0 \le \delta_1 + \delta_2 + \ldots + \delta_k - ky \le k-1$$

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 (1 1), (1 2) may be stated as

$$\delta_1 + \delta_2 + \ldots + \delta_k \ge 1 \qquad (14)$$

and

$$\delta_1 + \delta_2 + \ldots + \delta_k = 1 \qquad (15)$$

which represent the inclusive and exclusive .OR. respectively of k: logical variables. Now consider the relations

$$\delta_1 + \delta_2 + \ldots + \delta_k \ge P \qquad (16)$$

and

$$\delta_1 + \delta_2 + \ldots + \delta_k = p \qquad (17)$$

where p is an integer and $1 \le p \le k$. 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

$$\begin{aligned} R_1 &= \{(x_1 \ldots x_n) \mid \sum_{j=1}^{n} a_{1j} x_j \le b_1\} \\ &\vdots \\ R_m &= \{(x_1 \ldots x_n) \mid \sum_{j=1}^{n} a_{mj} x_j \le b_m\} \end{aligned} \qquad (18)$$

the constraint set

$$R = R_1 \ \& \ R_2 \ \& \ \ldots \ R_m \tag{19}$$

is stated as

$$R = \{(x_1 \ldots\ldots x_n) \mid \sum_{j=1}^{n} a_{ij}x_j \leq b_i,\ i = 1,\ldots\ldots,\ m\} \tag{20}$$

It may be observed that R, the intersection of $R_1, R_2, \ldots R_m$, is convex as R_i, $i = 1,\ldots,m$ 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

$$D = \{(x_1 \ldots x_n) \mid \ell_j \leq x_j \leq u_j,\ j = 1,\ldots,n\} \tag{21}$$

where some or all ℓ_j, u_j $j = 1,\ldots n$ are finite such that the bounds U_i, i=l...m are finite. Also from the redundancy consideration it is required that $b_i < U_I$, i=l,..,m.

To represent the inclusive .OR. relation

$$R_1 \ V \ R_2 \ V \ \ldots \ R_m \tag{22}$$

introduce the relations

$$\sum_{j=1}^{n} a_{ij}x_j - B_i(1 - \delta_i) \leq b_i,\ i = 1,\ldots\ldots\ m. \tag{23}$$

and

$$\sum_{i=1}^{m} \delta_i \geq 1. \tag{24}$$

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

$$F_i = \sum_{j=1}^{n} a_{ij}x_j .$$

Thus any finite value;' for B_i such that

$$B_i + b_i \geq U_i, \qquad i = 1,\ldots,m, \tag{25}$$

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

$$\sum_{i=1}^{m} \delta_i = 1 , \tag{26}$$

$$\sum_{i=1}^{m} \delta_i \geq p , \tag{27}$$

$$\sum_{i=1}^{m} \delta_i = p , \tag{28}$$

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

$$\begin{aligned} \text{Let } R_1 &= \{(x_1,x_2) \mid x_1 + x_2 \leq 4\} \\ R_2 &= \{(x_1,x_2) \mid -x_1 + x_2 \leq 0\} \\ R_3 &= ((x_1,x_2) \mid 3x_1 - x_2 \leq 8\} \end{aligned} \tag{29}$$

and let $D = \{(x_1,x_2) \mid 0 \leq x_1 \leq 5, 0 \leq x_2 \leq 5\}$

Then

$S = R\&D = R_1 \; \& \; R_2 \; \& \; R_3 \; \& \; D$ is as shown in Diagram 1 .

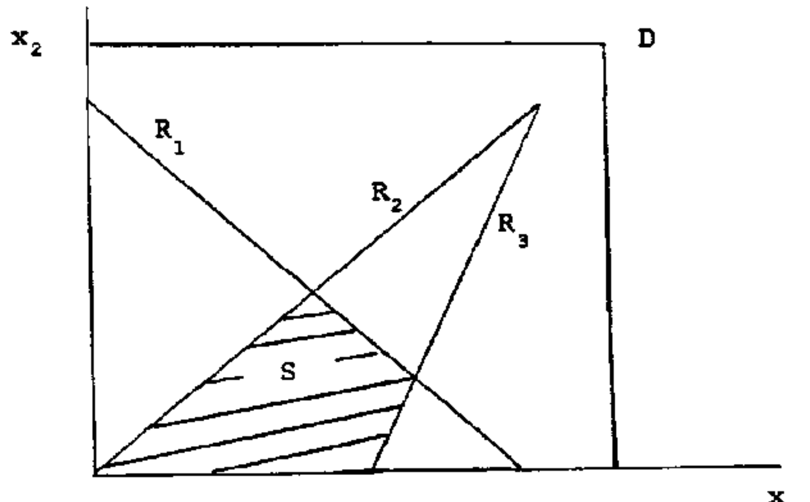

Diagram 1.

The three bounds on the linear forms may be computed as

$U_1 = 10 \quad , \qquad U_2 = 5, \qquad U_3 = 15$

A formulation which uses the logical .OR. as well as .AND. relations such as $T = R_1 V (R_2 \& R_3)$ may be stated as

$$
\begin{aligned}
x_1 + x_2 - 6(1-\delta_1) &\le 4 , \\
-x_1 + x_2 - 5(1-\delta_2) &\le 0 , \\
3x_1 - x_2 - 7(1-\delta 2) &\le 8 , \qquad (30) \\
\delta_1 + \delta_2 &\ge 1
\end{aligned}
$$

and $\delta_1 , \delta_2 = 0 , 1$.

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

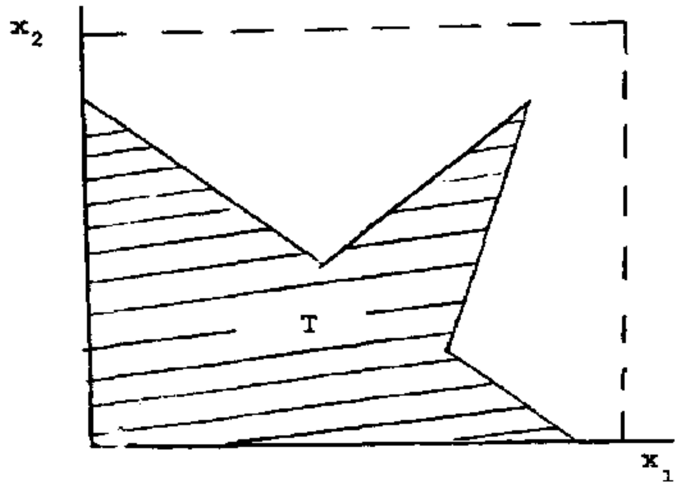

Diagram 2.

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

$$\text{Max} \sum_{j=1}^{n} f_j(x_j)$$

$$\text{subject to} \sum_{j=1}^{n} g_{ij}(x_j) \leq b_i, \qquad i=1,\ldots\ldots\ldots..m, \tag{31}$$

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 $f_j(x_j)$, $j = 1,..,n$ (32)

are all single valued.

(ii) The arguments x_j, $j = 1,\ldots,n$ of these functions have finite ranges ($\ell_j \leq x_j \leq u_j$, $j = 1,\ldots,n$) -

The construction of piecewise linear approximations 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

$$
\begin{array}{llll}
 & \text{Maximise} \quad f(x_1,\ldots,x_n) & & \\
\text{subject to} & g_i(x_1\ldots,x_n) \leq b_i\,, & i=1,\ldots,m & \\
 & g_i(x_1\ldots\ldots, x_n) = b_i\,, & i = m_i+1,\ldots,m & (33)\\
\text{and} & \ell_j \leq x_j \leq u_j\,, & j = 1,\ldots,n\,. &
\end{array}
$$

The following types of objective functions f(x) and constraint functions $g_i(x)$ are found in the set of problems.

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

$$f(x) = a_0 + \sum_{i=1}^{n} a_i x_i + \sum_{i,j=1}^{n} a_{ij} x_i x_j + \sum_{i,j,k=1}^{n} a_{ijk} x_i x_j x_k + \ldots\ldots \tag{34}$$

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

$$f(x) = \sum_{j \in J} c_j \underset{i}{\pi} x_i^{dij} \tag{35}$$

where J is used to label the terms appearing in the signomial function. In (34) a_0 ,a_i, a_{ij} etc, and in (35) c_j, d_{ij} 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						
		C	L	Q	S	P	G	Row sum
Constraint Function Codes	U							
	B			1	1	5	2	9
	L			10		8	6	24
	Q	1	7	18	2	9	1	38
	P		2	2		14	3	21
	G		3	6		7	7	23
	Column Sum	1	12	37	3	43	19	115

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 C, L or S and constraints with codes U, B or L are clearly in variable separable form.

<u>Product Term</u>

A product term, x_1x_2, may be replaced by $(y_1^2 - y_2^2)$ with the additional constraints $y_1 = ½(x_1 + x_2)$ and $y_2 = ½(x_1 - x_2)$. If $(\ell_i \le x= \le u_i)$ then, given finite ℓ_i and u_i, finite bounds L_i and U_i may easily be derived such that $(L_i \le y_i \le U_i)$, $i = 1,2$.

By repeated application of this technique a variable separable formulation of a higher order product term may be obtained.

<u>Quadratic Functionϕ</u>

For a general quadratic function, $\phi\ (x_1,\ldots x_n)$ a more compact variable separable formulation may be obtained.

Let $\phi(x_1,\ldots\ldots x_n) = \frac{1}{2}\sum\limits_{i=1}^{n}\ \sum\limits_{j=1}^{n} q_{ij}x_ix_j$ (36)

Replace $\phi(x_1,\ldots\ldots x_n)$ by $\psi(y_1,\ldots\ldots y_r) = \sum\limits_{k=1}^{r} d_k y_k^2$

With the constraints

$$y_k = \sum_{j=1}^{n} q'_k x_j \qquad k = 1,\ldots\ldots r \tag{37}$$

where r is the rank of the symmetric matrix $Q = \{q_{ij}\}$.

The coefficients q_{kj}'. and d_k can be determined by applying a standard method such as Lagrange's Reduction.

Given finite bounds ℓ_j and u_j on x_j, $j = 1,\ldots n$, 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 $H' = \sum\limits_{j=1}^{n} h'_j x_j$ and $H'' = \sum\limits_{j=1}^{n} h''_j x_j$.

The expression (H'/H") may be manipulated in the following way. Replaced (H′/H′′) by y_1 and introduce the constraint $\sum_{j=1}^{n} h'_j x_j = \sum_{j=1}^{n} h''x_j y_1$.

As discussed earlier a variable separable formulation may be obtained for the product terms of the constraint. The finite bounds on $x_j, j = 1,\ldots n$, provide bounds on H' and H" such that $L' \leq H' \leq U'$ and $L'' \leq H'' \leq U''$ from which bounds on y_1 may be obtained. If $L'' > 0$ or $U'' < 0$, the bounds on y_1 are finite and a piecewise linear formulation can be applied.

Power Forms - Constant Base

Consider the term $a^{x_1 + x_2^2}$ where $a > 0$.

A variable separable formulation may be obtained by replacing $a^{x_1 + x_2^2}$ by y_1 And introducing the constraint $\log y_1 = (\log a) . (x_1 + x_2^2)$. 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 $x_1^{x_2}$.This term can be handled using the substitution

$$y_1 = 10^{yx2} \tag{38}$$
$$x_1 = 10^{y2} \tag{39}$$

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 $0 < \ell_1 \leq x_1 \leq u_1$ from which the bounds on 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 $x_1 + 2x_2 + x_3$

Subject
$$x_1 x_2 + \frac{x_2}{1 + x_1} e^{x_3} + x_3 \leq 20 \tag{40}$$
$$x_1 + x_2 + x_3 \leq 4 \tag{41}$$
and
$$x_1, x_2, x_3 \geq 0. \tag{42}$$

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

$$0 \leq x_1, x_2, x_3 \leq 4$$

Rewrite
$$\frac{x_2}{1+x_1} = x_4 \text{ or } x_2 - x_4 - x_1 x_4 = 0 \tag{44}$$

Now from (43) and (44) $\ell_4 \leq x_4 \leq u_4$ where

$$\ell_4 = 0 \quad , \quad u_4 = 4 \tag{45}$$

The constraint (40) can be reexpressed as

$$x_1 x_2 + x_4 y_1 + x_3 \leq 20 \tag{46}$$

and $$y_1 = e^{x} 3 \tag{47}$$

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

$$e^{\circ} = 1 \leq y_1 < e^4 = 54.598$$

Thus the given problem may be restated as
Maximize $x_1 + 2x_2 + x_3$
subject to

$$\begin{aligned} x_1x_2 + x_4y_1 + x_3 &\leq 20 \\ x_2 - x_4 - x_1x_4 &= 0 \\ y_1 - e^{x3} &= 0 \\ x_1 + x_2 + x_4 &\leq 4 \end{aligned} \tag{48}$$

and $0 < x_1, x_2, x_3, x_4 \leq 4, \quad 1.0 \leq y_1 \leq 54.598$

The product terms are thus re-expressed as

$$x_1 x_2 = z_1^2 - z_2^2, x_4 y_1 = z_3^2 - z_4^2, x_1 x_4 = z_5^2 - z_6^2$$

which leads to the full separable programming formulation:
Maximise $x_1 + 2x_2 + x_3$

Subject to

$$\begin{aligned} z_1^2 - z_2^2 + z_3^2 - z_4^2 + x_3 &\leq 20 \\ x_2 - x_4 - z_5^2 + z_6^2 &= 0 \\ y_1 - e^{x_3} &= 0 \\ x_1 + x_2 + x_4 &\leq 4 \\ \tfrac{1}{2}x_1 + \tfrac{1}{2}x_2 - z_1 &= 0 \\ \tfrac{1}{2}x_1 + \tfrac{1}{2}x_2 - z_2 &= 0 \\ \tfrac{1}{2}x_4 + \tfrac{1}{2}y_1 - z_3 &= 0 \\ \tfrac{1}{2}x_4 + \tfrac{1}{2}y_1 - z_4 &= 0 \\ \tfrac{1}{2}x_1 + \tfrac{1}{2}x_4 - z_5 &= 0 \\ \tfrac{1}{2}x_1 + \tfrac{1}{2}x_4 - z_6 &= 0 \end{aligned}$$

$$0 \underset{z}{\leq} x_1 . x_2 . x_3 . x_4 \leq 4, \quad 1 \leq y_1 \leq 54.598$$

with $\ell_i^z \leq z_i \leq u_i^z$ as the easily derived bounds on the z_i, $i = 1, \ldots\ldots.6.$

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 International, 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 Practice, 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, Mathematical Programming, Vol. 10, pp 192-213, 1976.