TR/01/86 January 1986

LINEAR, INTEGER SEPARABLE AND FUZZY PROGRAMMING PROBLEMS: A UNIFIED APPROACH TOWARDS AUTOMATIC REFORMULATION

by

K. Darby-Dowman, C. Lucas

J. Yadegar and G. Mitra

Presented to: 27th Annual Conference of the Operational Research Society, Durham, 10-13 September 1985

LINEAR, INTEGER SEPARABLE AND FUZZY PROGRAMMING PROBLEMS: A UNIFIED APPROACH TOWARDS AUTOMATIC REFORMULATION

ABSTRACT

For mathematical programming (MP) to have greater impact as a decision tool, 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 after suitable algebraic manipulation put in a variable separable form. The methods of reformulating the fuzzy linear programming problem as a Max-Min problem is also introduced. It is shown that analysis of bounds plays a key role in the following four important contexts: model reduction, reformulation of logical restrictions as 0-1 mixed integer programs, reformulation of nonlinear programs as variable separable programs and reformulation of fuzzy linear 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 model reformulation techniques described here.

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 specialising 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], GAMS [13] have been developed. Fourer [l 4] for instance has given a very good review which sets out the scene as it stood by the end of 1979-80. Since then a number of other modelling systems have been reported LOGS [15], MAGIC [16], ULP [17]. 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.

Recently major issues covering the broad question of computer assisted modelling have received considerable attention. For instance in Greenberg and Maybee's [18] volume discussions and debates covering model structure, model simplification and solution analysis are presented. Palmer et al [19], [20] have discussed the design and the operational experience of an integrated model management system called PLATOFORM. Geoffrion [21] has put forward detailed analysis of the conceptual and theoretical aspects of data management and model structuring. For a brief review of recent developments the readers should refer to [22].
We have designed and implemented an interactive modelling system: the design objectives of this computer assisted mathematical programming (modelling) system (CAMPS) are set out in [22] and [23]. Details of the supporting optimiser may be found in [24]. Within this integrated system it is possible to construct, solve, analyse and document linear programming problems.
The modelling support discussed so far assumes that a model has already been conceived and a mathematic formulation using suitable symbols has been set out. These systems then serve to communicate the model to the computer based optimiser and to analyse the solution in relation to the original model.
It is well known that reformulations of integer, and variable separable programming problems also require considerable insight and modeling skill. Our experience with users of modelling support systems has convinced us that there is a great scope for providing automatic support for reformulating such nonlinear programming problems. The purpose this paper is to present a unified approach towards a range of such problems. Thus the methods described here can fit naturally into most LP modelling support systems.

The contents of this paper are organised as follows. In section 2 the LP is defined in a general form and this is mainly to introduce notation which is used in the rest of the paper. Analysis of bounds for linear forms is well known in the context of model reduction [25], [26]. Some of the bound analysis results which are pertinent to model reformulation as well are presented in section 3. The principles and methods underlying the reformulation technique are described in section 4. The main emphasis of this section is to show how logical statements (clausal forms) can always be restated as equivalent integer forms involving 0-1 integer variables. Strategies for separating variables to represent a wide range of nonlinear programming problems are presented and discussed in section 5. Reformulation of fuzzy programming problem as a max-min LP problem and the relationship of this approach to IP reformulation methods are presented in section 6. The general scope and applicability of these reformulation methods are discussed in section 7. The contents of sections 3, 4, 5, 6 have been gathered from diverse and independent sources and as such do not contain new material. However, this paper provides a general unification of otherwise independent methods. Thus in our analysis and automatic reformulation strategy we present a different focus on the underlying modelling principles and structure of these problems.

2. Statement of the General LP Problem and Notation

The notation introduced in this section is used in the rest of the paper. We state the general LP problem in the following form:

- Subscripts and their ranges

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

- Variables, constraints, and matrix coefficients:

  $x : x_j, \ j = 1 \ldots n, \ r : r_i, \ i = 1 \ldots m, \ d : d_j, \ j = 1, \ldots n,$

  $c : c_j, \ j = 1 \ldots n, \ b : b \ , \ i = 1 \ldots m,$

  $A : a_{ij}, \ i = 1 \ldots m, \ j = 1, \ldots n.$

- Linear objective function and constraints:

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

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

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

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

where ℓ_j may be $-\infty$ or finite and u_j may be $+\infty$ or finite.

3. Analys is 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 (ı) and discussed in section 2. Express these as two sets R and D of Linear Form constraints and Structural constraints respectively.

$$R = \{(x_2, \ldots x_n) \mid \sum_{j=1}^{n} a_{ij}x_j p_i b_i, \ i=1,\ldots,m \tag{2}$$

$$D = \{(x_2, \ldots x_n) \mid \ell_j \le x_j \le u_j, j=1,\ldots n\} \tag{3}$$

It is well known [25],[26], 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 redundant,

(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

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

denote the ith 1inear form.

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

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

Let $L_i \le F_i \le U_i, \quad i=1,\ldots,m$ (6)

denote the bounds on the linear form F_i; then from the definition of the structural bounds ($l_j \le x_j \le U_j$)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 [25].

3.3 Examples

Example 1 A Redundant Constraint

Let the constraint sets R and D be as defined below.

$$R=\{(x_1,x_2,x_3) \mid x_1 + 2x_2 - x_3 \leq 11\} \tag{9}$$

$$D=\{(x_1,x_2,x_3) \mid 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 2, 0 \leq x_3 \leq 4\}$$

The bounds on the linear form F_1 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

$$R = \{(x_1,x_2,x_3) \mid x_1 + x_2 - 2x_3 = 2\} \tag{10}$$

$$D =\{(x_1, x_2, x_3) \mid 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 2, 0 \leq x_3 \leq 4\}$$

Since $a_{13} < 0$ and ρ_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, nto variable separable functions with arguments defined between upper and lover bounds.

(v) In the reformulation of fuzzy programs as crisp max-min linear programs the upper bound values U_i may be used to check the consistency of the membership function.

The consequences of these observations in relation to reformulation methods for integer, separable and fuzzy programming problems 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 . We 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 [27], [3], [28].

Recently Jeroslow et al [29] have set out an exposition and also present experimental results which connect integer programming with propositional logic and theorem proving. They, for instance, consider three well known clausal forms conjunctive normal form, disjunctive normal form and Horn sentence. They then show how the equivalent integer forms may be constructed. Our interest, of course, is to interpret such theory and develop reformulation techniques for integer programming.

In this section we first introduce the necessary notation.
Let

Δ_i i = 1,2,... denote logical variables which may take value. TRUE. or .FALSE., and
δ_i i = 1,2,... 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 '

Representing .OR.

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

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

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 $\Delta_1 V\Delta_2 V\Delta_3 \ldots \Delta_k$ is .TRUE. (which is expressed as $Y \equiv \Delta_1 V\Delta_2 V\Delta_3 \ldots \Delta_k$), can be represented by the constraint

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

We note that (11) as set out above is an Integer form representation of the disjunctive normal form clause.

Representing .AND.

The logical condition

$$Y \equiv (\Delta_1 \,\&\, \Delta_2 \,\&\, \ldots \,\&\, \Delta_k) \tag{14}$$

can be represented by the constraint

$$0 \leq \delta_1 + \delta_2 + \ldots \delta_k - ky \leq k - 1 \tag{15}$$

p - fold Alternatives

The general forms of the relations (11),(12) may be stated as

$$\delta_1 + \delta_2 + \ldots \delta_k \geq 1 \tag{16}$$

and

$$\delta_1 + \delta_2 + \ldots \delta_k = 1$$

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

$$\begin{aligned} &\delta_1 + \delta_2 + \ldots \delta_k \geq p \\ \text{and} \quad &\\ &\delta_1 + \delta_2 + \ldots \delta_k = p \end{aligned} \tag{17}$$

where p is an integer and $1 \leq p < k$. The relations in (17) represent the statement Mp or more alternatives hold at any time" and 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 En. If a number of these are stated and need to be satisfied then these invoke the logical .AND. operation.

Thus for

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

Let P denote the proposition that $x \in R$, where

$$R = \{ (x_1 \ldots x_n) \mid \sum_{j=1}^{n} a_{ij} x_j \leq b_i, \; i = 1 \ldots m \} \tag{19}$$

and let P_i denote the proposition that

$x \in R_i$, $i = 1 \ldots m$. Then we note that

$$P = P_1 \; \& \; P_2 \; \& \; \ldots P_m \tag{20}$$

We observe 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 propositions $P_1, P_2, \ldots P_m$ it is necessary to consider the structural constraint set

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

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

To represent the inclusive .OR. relation

$$P_1 \; V \; P_2 \; V \; \ldots \; P_m \tag{22}$$

introduce the relations

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

and

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

where B_i. is a finite value such that for $\delta_i = 0$, $B_i + b_i$ is greater than or equal to the upper bound of

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

Thus any finite value, for B_i such that

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

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 [27] and modified.

$$\text{Let} \quad R_1 = \{(x_1, x_2) \mid x_1 + x_2 \leq 4\}$$

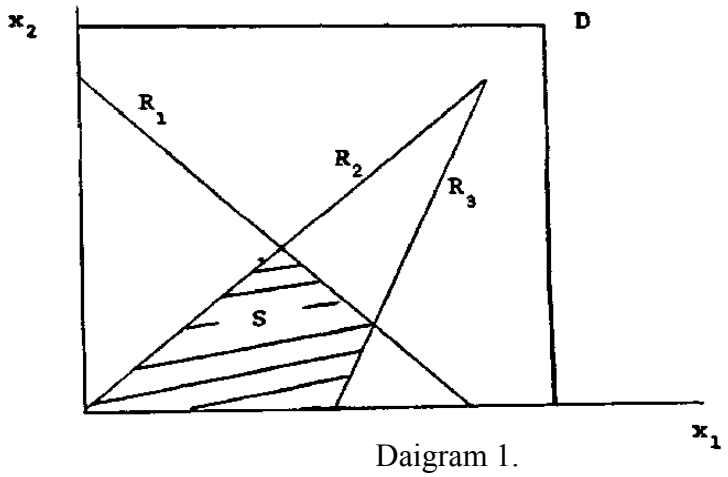
$$R2 = \{(x_1,x_2) \mid -x_1 + x_2 \leq 0\} \tag{29}$$

$$R3 = \{(x_1,x_2) \mid 3x_1 - x_2 \leq 8\}$$

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

The proposition that x satisfies all the constraints is implied in the LP formulation. Thus $x \in S$, where S is defined as

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



Daigram 1.

The three bounds on the linear forms may be computed as

$U_1 = 10$, $U2 = 5$, $U3 = 15$

A formulation which uses the logical .OR. as well as .AND. relations is now set out. The proposition

$$P_1 \, v(P_2 \,\&\, P_3)$$

is equivalent to: "find x which satisfies R_1 or R_2 and R_3". Introduce $\delta_1 = 0,1$ for P_1 and $\delta_2 = 0,1$ for $(P_2 \& P_3)$. We need to define the restrictions to represent T where $T = R_1 \, V(R_2 \,\&R_3)$. The equivalent mixed integer linear programming formulation may be stated as,

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

and

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

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

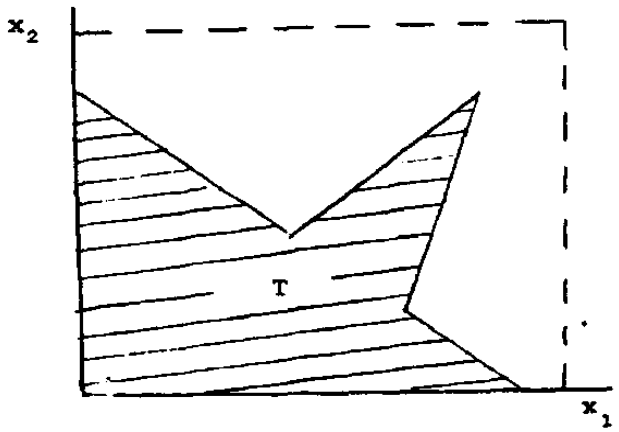


Diagram 2 .

Because of the inclusive .OR, relation the constraint set T which is constructed with a union operation and represented by the mixed integer formulation (30) is not a convex set.

5. Strategies for Separating Variables in Non Linear Programming Problems

5.1 Linearization of Variable Separable Programming Problems.

The problem

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

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

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 $f_j(x_j)$, $j = 1, \ldots, n$ (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 and the methods of solution are well discussed in [27], [30], [31] , [32] , [33].

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 we have analysed the comprehensive collection of nonlinear programming test problems which have been put together in [34].

Consider the test problems in the format

$$\begin{array}{lll} \text{Maximise} & f(x_1 \ldots, x_n) & \\ \text{subject to} & g_i(x_1 \ldots, x_n) \leq b_i \ , & i = 1, \ldots, m_1 \\ & g_i(x_1 \ldots, x_n) = b_i \ , & i = m_1 + 1, \ldots, m \\ \text{and} & \ell_j \leq x_j \leq u_j \ , & j = 1, \ldots, n. \end{array} \tag{33}$$

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

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

It may be observed that in geometric programming problem [35] a more general formis introduced which is called the signomial function and is expressed as

$$f(x) = \sum_{j \in J} c_j \prod_{j} x_i^{d_{ij}} \qquad (35)$$

where J is used to label the terms appearing in the signomial function. In (34) a_o ,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 [34] 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 variable which may appear in the problem.

We note that McCormick and Jackson [36] have done considerable work on the (reformulation) factorization of highly complex nonlinear programming problems. They analytically derive the hessian and gradient of the 'factored' forms and are interested in the sensitivity properties of the resulting nonlinear models.

In this section we consider a few frequently occuring instances of nonlinearity (nonlinear terms as well as nonlinear forms) and briefly discuss methods of reformulating these.

Product Term

A product term, x_1x_2, may be replaced by $(y_1^2 - y_2^2)$ with the additional constraints $y_1 = \frac{1}{2}(x_1 + X_2)$ and $y_2 = \frac{1}{2}(x_1 - X_2)$. If $(\ell_i \leq x_i \leq u_j)$ then, given finite ℓ_i and u_i, finite bounds L_i and U_i may easily be derived such that $(L_i \leq y_i \leq U_i)$, i = 1,2.

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

Quadratic Function

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

$$\text{Let } \varphi\ (x_1,\ldots x_n) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} q_{ij}x_i x_j \tag{36}$$

$$\text{Replace } \varphi\ (x_1,\ldots x_n) \text{ by } \psi\ (y_1,\ldots y_r) = \sum_{k=1}^{r} d_k y_k^2$$

with the constraints

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

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 Gaussian reduction [36].

Given finite bounds ℓ_i and u_j on x_j, j = 1,...n, finite bounds L_k and U_k on y_k, k = 1,...r, may be simply derived by considering the linear forms (37), thus enabling a piecewise linear approximation to be used.

Ratio of Linear Forms

$$\text{Let } H' = \sum_{j=1}^{n} h'_j x_j \text{ and } H'' = \sum_{j=1}^{n} h''_j x_j .$$

The expression (H'/H") may be manipulated in the following way.

Replace (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,...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 $L^{\Pi} > 0$ 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 y_1 can be derived from the bounds on x_1 and x_2.

Power Forms - Variable Base

Consider the term $x_1^{x_2}$. This term can be handled using the substitution $y_1 = x_1^{x_2}$ and introducing the constraints

$$y_1 = 10^{y_2 x_2} \tag{38}$$

$$x_1 = 10^{y_2} \tag{39}$$

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 be immediately formulated as variable separable lie in the class in which the objective or constraint code is G. However, most practical problems in this class can be transformed to a separable form without difficulty. To illustrate this point an example is set out below.

5.4 An Example

Consider the problem [38].

Maximise $x_1 + 2x_2 + x_3$

subject to $$x_1 x_2 + \frac{x_2}{1+x_1} e^{x_3} + x_3 \leq 20 \tag{40}$$

$$x_1 + x_2 + x_3 \leq 4 \tag{41}$$

and $$x_1, x_2, x_3 \geq 0. \tag{42}$$

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

$$0 \leq x_1, x_2, x_3 \leq 4 \tag{43}$$

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) $\ell_4 \le x_4 \le u_4$ where

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

The constraint (40) can be reexpressed as

$$x_1x_2 + x_4y_1 + x_3 \le 20 \tag{46}$$

and $$y_1 = e^{x_3} \tag{47}$$

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

$$e^0 = 1 \le y_1 \le e^4 = 54.598$$

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

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

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

The product terms are thus re-expressed as

$$x_1x_2 = z_1^2 - z_2^2 \,, x_4y_1 = z_3^2 - z_4^2, x_1x_4 = z_5^2 - z_6^2$$

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

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

$$0 \le x_1, x_2, x_3, x_4 \le 4, \qquad 1 \le y_1 \le 54.598$$

with $\ell^z_i \le z_i \le u_i$ as the easily derived bounds on the z., i = 1,...6.

6. Reformulation of Fuzzy Decision Problems as Max-Min LP Problems

6.1 Background to the Model

Fuzzy set theory was first introduced by Zadeh [39] and subsequently Bellman and Zadeh [40] discussed its application to decision problems. Later developments and applications of this approach are well discussed in the text book by Didier and Dubois [41] . In Fuzzy set theory an element x is defined to have a degree o£ membership of a given set say S. The degree of membership is denoted by a membership function $\mu(x)$ which is defined over the range [o,u] where u is a positive real number. For u = l we have a normal fuzzy set, $\mu(x) \in [0,1]$. In the usual set theoretic terms x belongs to S is equivelent to $\mu(x) = 1$ and $\mu(x) = 0$ otherwise.

The major contribution of the seminal paper by BelIman and Zadeh [40] was to establish the relationship between goals and constraints of a decision problem. In their words:

> "goals and the constraints constitute classes of alternatives whose boundaries are not sharply defined." They then proceed to explain that their modelling framework..." erases the differences between goals and constraints and makes it possible to relate in a relatively simple way the concept of a decision to those of the goals and constraints of a decision process... In short, a broad definition of the concept of decision may be states as:
>
> Decision = Confluence of Goals and Constraints".

Fuzzy Programming as a decision model was mainly promoted by Zimmermann [42]. Its applications to media selection [43], and power systems planning [44] are two of many applications which have been reported. Dyson [45] considers the multicriteria decision problems, analyses it following the Max-Minapproach based on utility function and shows how the latter has the identical form to that of crisp equivalent formulation of the fuzzy LP.

6.2 Statement and Reformulation of Fuzzy Linear Programs

Consider the linear programming problem with 1,...k, objective (goal) functions and m inexact (soft) restrictions defined as

$$\text{Max } Z = C_x \tag{49}$$

$$\text{subject to} \quad Qx \lesssim d \tag{50}$$
$$x \geq 0$$

where d is an m vector
C is a k × n matrix
Q is an m × n matrix

Let $\bar{z} = \begin{pmatrix} z_i \\ z_k \end{pmatrix}$ denote the 'aspiration levels' (that is the maximum these are expected to achieve) of these k objectives. Define

$A = \begin{bmatrix} C \\ Q \end{bmatrix}$ a (k+m) × (n) matrix, $b = \begin{bmatrix} \bar{z} \\ d \end{bmatrix}$ a (k+m) vector.

Let
$$\mu_i(x) = f_i\left(\sum_{j=1}^{n} a_{ij} x_j\right) \tag{51}$$

denote the membership function of the ith goal or restriction, i=1 ,2. . .k+m.

A typical membership function is illustrated in Diagram 3

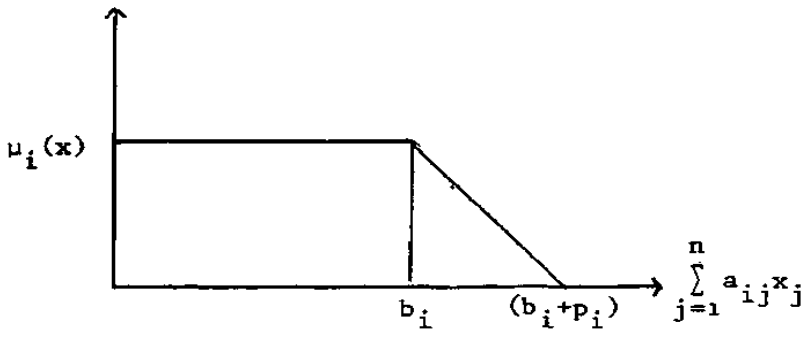


Diagram 3

Thus we define

$$\mu_i(x)=\begin{cases} 1 & \text{if } \sum_{j=1}^{n} a_{ij}x_j \le b_i \\ 1=\dfrac{\left(\sum_{j=1}^{n} a_{ij}x_j - b_j\right)}{P_i} & \text{if } b_i < \sum_{j=1}^{n} a_{ij}x_j \le (b_i+p_i) \\ 0 & \text{if } \sum_{j=1}^{n} a_{ij}x_j > b_i + p_i \ . \end{cases} \tag{52}$$

If μ_d (x) denotes the membership function of the (optimal) decision set then following the usual (but much debated) approach of applying 'Min' as the intersection operator we have

$$\mu_D(x) = \min_i \mu_i(x) \tag{53}$$

Thus maximum satisfaction of constraints and targets are achieved by solving the equivalent Max-Min linear program,

Max λ

$$\text{subject to } \lambda p_i + \sum_{j=1}^{n} a_{ij}x_j \le b_i + p_i \ , \quad i = 1...k+m, \tag{54}$$
$$x_j \ge 0, \ j = 1 \ldots n \ .$$

We can make following observations for this model.

(a) The multiple objective (or goal) model illustrates Zadeh and Bellman's principle (see section 6.1) rather well. In case of single objective function we have k = 1 .

(b) The fuzzy goals and constraints are alternative ways of introducing soft constraints in the model.

(c) If we have the variables x_j bounded, that is . $\ell_j \le x_j \le u_j$ as in section 3, then U_i as introduced in that section may be used to check the consistency of the fuzzy membership function. Clearly $b_i + p_i \le U_i$.

(d) If we wish to construct models which involve crisp as well as fuzzy relations then reformulation methods of section 4 and section 6 can be naturally put together.

7. Discussions

In this paper we make a case for integrating and automating a number of reformulation methods of mathematical programming. We also illustrate the key role played by bound analysis in these methods. Currently most modelling support systems only allow the user to create the underlying LP model. Here we show how this basic modelling tool can be naturally extended to incorporate reformulation support. By introducing the facility of algebraic manipulation it is possible to reduce the chore of manual reformulation of models. This aspect may prove to be particularly valuable for problem owners who are capable of describing their problems precisely but may not be experienced in reformulation techniques. 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. Bisschop, J., and Meeraus, A., On the Development of a General Algebraic Modelling System in a Strategic Planning Environment, Mathematical Programming Study 20, 1982, North Holland.
14. Fourer, R., Modelling Languages versus Matrix Generators for Linear Preogramming, ACM Transactions on Mathematical Software, vol. 9, No.2, 143-183, 1983.
15. Brown, R.W., Northup, W.D., and Shapiro, J.F., LOGS: A Modelling and Optimization System for Business Planning, in Computer Assisted Decision Making, Editor G. Mitra, North Holland, 1985.
16. Williams, H.P., Matrix Generator Input Convertor: MAGIC, University of Edinburgh, report, 1983.
17. Witzgall, c., and McClain, M. , Problem and Data Specification for Linear Programs, U.S. Department of Commerce, National Bureau of Standards, Report NBSIR 85-3125, April, 1985.
18. Greenberg, H.J., and Maybee, J.S., Computer Assisted Analysis and Model Simplification, Academic Press, 1981.
19. Palmer, K.H. et al, A Model Management Framework for Mathematical Programming, An Exxon Monograph, John Wiley, New York, 1984.
20. Rowland, A.J., A Data and Model Management System in Exxon, in Computer Assisted Decision Making , see reference 15 above.

21. Geoffrion, A.M., Structured Modelling, Western Management Science Institute, Graduate School of Management, University of California, Los Angeles, CA 90024, 1985.

22. Lucas, C. Mitra, G., and Darby-Dowman, K., Modelling of Mathematical Programs: An Analysis of Strategy and Proposal for a Computer Assisted System, TR/09/83, Brunel University, revised January 1986.

23. Lucas, C., and Mitra, G., Computer Assisted Mathematical Programming (Modelling) System: CAMPS, User Reference Manual, Brunel University, July, 1985.

24. Mitra, G., Tamiz, M., and Yadegar, J., A Suite of FORTRAN based LP subroutines, FORTLP, User Specification, Brunel University, Dept. of Mathematics and Statistics, July 1985.

25. 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.

26. 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.

27. Williams, H.P., Model Building in Mathematical Programming, John Wiley & Sons, 1978.

28. Simonard, M., Linear Programming, Prentice Hall International, 1966.

29. Blair, C.E., Jeroslow, R.G., and Lowe, J.K., Some Results and Experiments in Programming Techniques for Propositional Logic, Report of Georgia Institute of Technology, January 1985.

30. Mitra, G., Theory and Application of Mathematical Programming, Academic Press, 1976.

31. Bradley, S.P., Hax, A.C., and Magnanti, T.L., Applied Mathematical Programming, Addison-Wesley, 1977.

32. Beale, E.M.L., Mathematical Programming in Practice, Pitman 1968.

33. Hadley, G.H. Nonlinear Programming, Addison Wesley, 1964.

34. Hock, W., and Schittkowski, K. Text Examples for Nonlinear Programming Codes, Springer Verlag, 1981.

35. Dembo, R.S., A set of Geometric Programming Test Problems and their Solutions, Mathematical Programming, Vol.10, pp 192-213, 1976.

36. Jackson, R.H.F., and McCormick, G.P., The Polyadic Structure of Factorable Function Tensors with Application to High Order Minimisation Techniques, Dept of Operations Research, George Washington University, 1984.

37. Stiefel, E.L., An Introduction to Numerical Mathematics, Academic Press 1963.

38. Question 6, Honours paper in Mathematical Programming, Bachelor of Science Degree Examination, Mathematics, Statistics, Engineering, Brunel University, 1982.

39. Zadeh, L.A., Fuzzy Sets, Information and Control, Volume 8, p338-353, 1965;

40. Bellman, R., and Zadeh, L.A., Decision Making in a Fuzzy Environment, Man. Sci, Vol 17, pp 141-164, 1970.

41. Dubois, D., and Prade, H., Fuzzy Sets and Systems, Academic Press, New York, 1980.

42. Zimmermann, H.J. Fuzzy Programming and Linear Programming with Several Objective Functions, Int. Journal Fuzzy Sets Syst Vol. 1, pp 45-56, 1978.

43. Zimmermann, H.J., and Wiedey, G., Media Selection and Fuzzy Linear Programming, JORS, Vol 29, pp 1071-1084, 1978.

44. Satoh, H., and Serizawa, Y., An Application of Fuzzy Linear Programming to Expansion Planning of Electric Power Generation, IEE, Japan Technical Meeting, PE-82-3, 1982.

45. Dyson, R.G., Maximin Programming, Fuzzy Linear Programming and Multi-Criteria Decision Making, JORS, Vol 31, pp 263-267, 1980.