Mathematical Programming 1 (1971) 321–358. North-Holland Publishing Company

A PROJECTIVE METHOD FOR STRUCTURED NONLINEAR PROGRAMS [*], [**]

Michael D. GRIGORIADIS

IBM Corporation, New York, N.Y., U.S.A.

Received 9 November 1970
Revised manuscript received 27 May 1971

This paper describes a partitioning method for solving a class of structured nonlinear programming problems with block diagonal constraints and a few coupling variables.

The special structure of the constraints is used to reduce the given problem by elimination of variables. In variance to other methods proposed previously, this elimination is effected through the use of the general solution to an underdetermined system of linear equations representing the active constraints at a given feasible point. For weakly coupled systems, this arrangement provides a drastic reduction in the number of variables. The solution to the overall problem is obtained by solving a sequence of the reduced nonlinear programs. Primal feasibility is maintained throughout the optimization procedure. Computational experience and results are presented.

1. Introduction

Large mathematical programming models arising in practice almost always exhibit some structure in their constraints and in the composition of their objective function. The most common of these is the block diagonal structure with a few coupling constraints or variables or both. Such problems usually originate from industrial applications involving the scheduling of production, inventory and distribution activities. Other sources include dynamic economic systems, decentralized production systems and some stochastic programming problems.

[*] This paper was presented at the 7 th Mathematical Programming Symposium 1970, The Hague, The Netherlands.

[**] This work was supported in part by the National Science Foundation under Research Grant GJ-0362 and in part by the New York Scientific Center, IBM Corporation.

During the last decade, various methods of treating such large problems have been proposed. Geoffrion [1] gives excellent reviews of these methods and a complete compilation of references for this area. The Dantzig–Wolfe Decomposition Principle [2] was the first method suggested for solving structured linear programming problems. Treatment of structured nonlinear programs by Partition Programming was originally proposed by Rosen [3,4].

The partitioning method presented here applies to a class of nonlinear programs, stated in section 2, with linear constraints exhibiting the familiar block diagonal structure and a relatively small number of coupling variables. This class of "weakly coupled" problems differs from the one treated in [4] since, for fixed levels of the "coupling variables" y, the problem naturally decomposes into several *nonlinear* programming problems each in the "block variables" x_j; $j = 1, ..., k$. Elimination of variables based on basis arguments is not effective since normally, the number of active hyperplanes at a feasible point $(\bar{x}_1^t, ..., \bar{x}_k^t, y^t)$ is much smaller than the dimensionality of the problem. The proposed method uses the general solution to an underdetermined system of linear equations representing the active constraints at $(\bar{x}_1^t, ..., \bar{x}_k^t, y^t)$ to eliminate all x_j variables. The transformed problem, in terms of the coupling variables y, is then solved to obtain a step and an improved point $(x_1^{t+1}, ..., x_k^{t+1}, y^{t+1})$. Subsequent minimization in at least one block with fixed $y = y^{t+1}$ gives a further possible improvement in the x_j-spaces resulting in the point $(\bar{x}_1^{t+1}, ..., \bar{x}_k^{t+1}, y^{t+1})$ and a possible change in the set of active contraints for that block. The process is then repeated, starting with this improved point.

Geometrically, this general procedure is equivalent to limiting the search for a feasible point with an improved function value to points lying in the intersection of the currently active block constraints. This assumption allows the establishment of a mapping of feasible points onto the space of the coupling variables y and an inverse mapping of feasible points in the y-space onto the (temporarily fixed) intersection of active constraints. These mappings may be performed by projection operators associated with the "particular solution" to the generally underdetermined linear system of equations representing the active constraints in each block, assuming that the homogeneous solution is temporarily fixed. This assumption is then effectively relaxed by fixing y and minimizing in the x_j-spaces for some blocks j.

The algorithm may best be illustrated by a small example of a one block problem in $\mathbb{R}^3$ (fig. 1.1). The block variables are $x =$

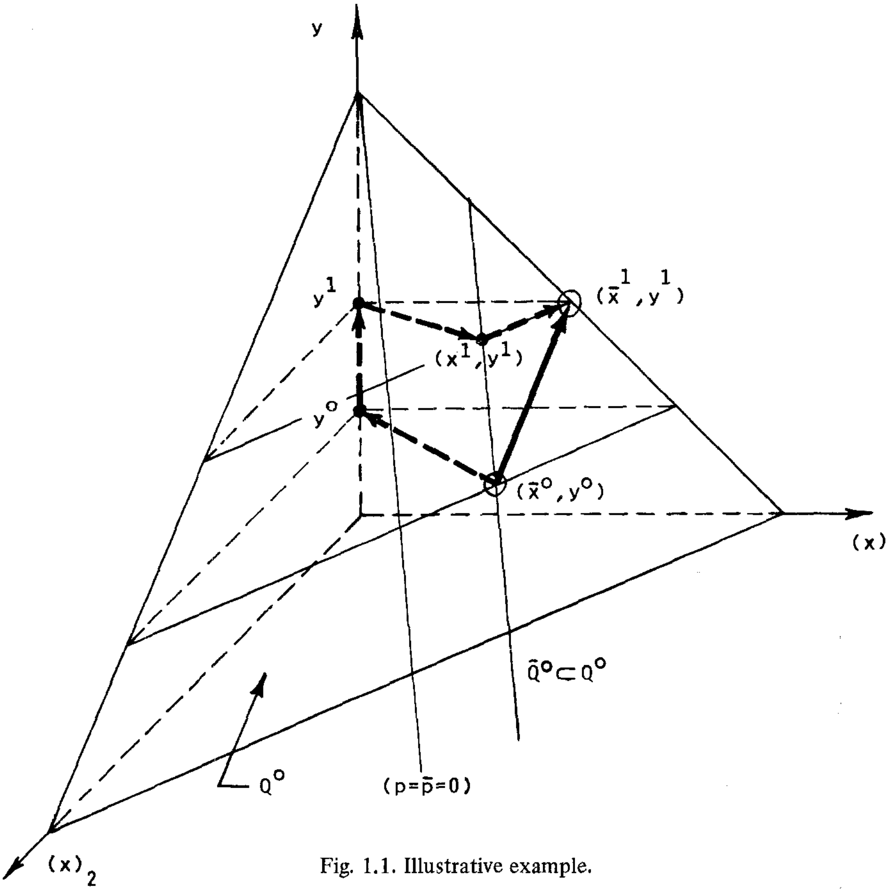

Fig. 1.1. Illustrative example.

$((x)_1, (x)_2) \in \mathbb{R}^2$ and the coupling variable $y \in \mathbb{R}^1$. The feasible region consists of the nonnegative orthant in $\mathbb{R}^3$ bounded by the plane denoted by Q^o. Suppose that a feasible point $(\bar{x}^o, y^o) \in Q^o$ is given. Elimination of the variables x using the linear system of equations defining Q^o, or the equivalent projection operation, projects the feasible region onto the y-space. The implication here is that the homogeneous solution has been fixed to some vector $p = \bar{p} \in \mathbb{R}^2$. It corresponds to temporarily limiting the search to points lying in the subspace $\bar{Q}^o \subseteq Q^o$ parallel to that denoted by the parameter $p = \bar{p}$. Now a minimization over feasible points in the y-space is performed to obtain an improved point y^1. The corresponding x^1 is obtained by projecting y^1 onto the manifold $\bar{Q}^\circ \subseteq Q^\circ$ defined by $p = \bar{p}$. When $\bar{p} = 0$, this may be regarded as choosing x^1 so that $(x^1, y^1) \in \bar{Q}^o \subseteq Q^o$ and so that, for the chosen y^1, $(x^1 - \bar{x}^o)$ has minimum Euclidean norm. The effect of fixing

$\bar{Q}^o \subseteq Q^o$ and Q^o is now reviewed and possibly rectified, by fixing $y = y^1$ and performing a minimization in the x-space. This results in the point $(\bar{x}^1, y^1)$ and the new set of active constraints given by the intersection of Q° and the subspace $x_1 = 0$. The process is repeated starting with $(\bar{x}^1, y^1)$.

Section 2 contains a formal statement of the structured nonlinear problem (P). Section 3 describes the theoretical and algorithmic aspects of the proposed method. The algorithm is presented in section 4 and the question of minimization within the manifold Q^t is examined in section 5. Some computational aspects and experimental results are given in section 6. *Appendix 1* presents some results for underdetermined systems of linear equations and *Appendix 2* gives a small numerical example.

2. The nonlinear problem (*P*)

Let $x_j \in \mathbb{R}^{n_j}$; $j = 1, ..., k$, $y \in \mathbb{R}^{n_o}$. For notational purposes let the pair $(x, y) \in \mathbb{R}^{n_o+n}$; $n = \sum_{j=1}^{k} n_j$ represent the vector $(x_1, ..., x_k, y) \in \mathbb{R}^{n_o+n}$. Let $\bar{S}_j$ be open sets in $\mathbb{R}^{n_j}$; $j = 0, 1, ..., k$, let Y be a closed convex subset of $\bar{S}_0$ in $\mathbb{R}^{n_o}$, and let the given numerical function

$$f(x, y) = f_0(y) + \sum_{j=1}^{k} f_j(x_j, y) \tag{2.1}$$

be continuously differentiable and convex over the open set $\bar{S} = \bar{S}_1 \times ... \times \bar{S}_k \times \bar{S}_0$. Furthermore, let h_j be given m_j-vectors, B_j be given (n_j, m_j)-matrices and D_j be given (n_j, m_j)-matrices for $j = 1, ..., k$ such that D_j is nonzero for at least one j. We let $m = \sum_{j=1}^{k} m_j$. The nonlinear minimization problem (P) to be considered here is then defined as:

P: Find (x^*, y^*), if it exists, such that

$$f(x^*, y^*) = \min \{f(x, y) \mid (x, y) \in S = \bigcap_{j=0}^{k} S_j\} \tag{2.2}$$

$$S_o = \{x_j \in \mathbb{R}^{n_j}; j = 1, \ldots, k, y \in Y\} \subseteq \mathbb{R}^{n_o+n} \tag{2.3}$$

$$S_j = \{x_j \in \bar{S}_j; y \in Y \mid B_j'x_j + D_j'y \leq h_j\} \subseteq \mathbb{R}^{n_o+n} \;;\; j = 1, \ldots, k\,. \tag{2.4}$$

It will be assumed that the sets S_j; $j = 1, \ldots, k$ are bounded for each $y \in Y$. Hence, S is compact convex and the differentiable function $f(x, y)$ will achieve a finite minimum on S.

In practice, the set Y will be either defined as $Y \equiv \mathrm{R}^{n_o}$ or specified in terms of the nonempty set

$$Y = \{y \in \bar{S}_o \mid D_o(y) \leq h_o\} \tag{2.5}$$

where $D_o(y)$ is a given m_o-dimensional continuously differentiable and convex vector function on $\bar{S}_o$ and h_o is a given m_o-vector. It will be assumed that Y satisfies the Kuhn–Tucker or Slater's constraint qualification (see e.g. Mangasarian [5]). However, quite commonly $D_o(y) \equiv D_o'y$ where D_o is a given (n_o, m_o)-matrix.

Procedures for handling linear equality constraints directly, without increasing the size of the problem and implicit treatment of given upper or lower variable bounds are given in [6].

In order to simplify the ensuing discussion we will consider P for only one block ($k = 1$) and omit the block subscript j whenever $j = 1$, except where necessary to avoid notational confusion. We will also use the multiblock notation whenever appropriate. The interior and boundary of a set A will be denoted by $\mathrm{Int}(A)$ and $Bdy(A)$ respectively. A prime will indicate the transpose of a matrix.

3. The method

Definition 3.1: A given $y = y^t$ will be called *feasible* if and only if $y^t \in Y$ and there exists a vector

$$x^t = x(y^t) \in X(y^t) = \{x \in \bar{S}_1 \mid B'x \leq h - D'y^t\}; X(y^t) \subset S_1. \; \square \tag{3.1}$$

In practice, a *feasible* y is usually known or can be constructed. For a fixed *feasible* $y = y^t$, P reduces to k smaller nonlinear programs

PI^t: Find $\overline{x}^t$, such that:

$$f(\overline{x}^t, y^t) = \min_x \{ f(x, y^t) \mid x \in X(y^t) \} \tag{3.2}$$

which will be referred to as *Problem I at* $y = y^t$ and generally denoted by $PI^t_j (\equiv PI_j(y^t))$; $j = 1, \ldots, k$.

Assuming that $\overline{x}^t \in Bdy(X(y^t))$, the *feasible* $(\overline{x}^t, y^t)$ induces the following partitioning of the constraints of P:

$$\underline{D}^t_o(y^t) = \underline{h}^t_o \qquad ; \overline{D}^t_o(y^t) < \overline{h}^t_o \tag{3.3}$$

$$\underline{B}^{t\prime}\overline{x}^t + \underline{D}^{t\prime} y^t = \underline{h}^t \qquad ; \overline{B}^{t\prime}\overline{x}^t + \overline{D}^{t\prime} y^t < \overline{h}^t \tag{3.4}$$

and the corresponding partitioning of Lagrange multipliers into:

$$u^t = (\underline{u}^t_o, \overline{u}^t_o, \underline{u}^t, \overline{u}^t). \tag{3.5}$$

where the underbar denotes the *active* and the overbar the *inactive* constraints. We let $I_o = \{1, \ldots, m_o\}$ be the index set of the constraint rows defining Y; $I = \{1, \ldots, m\}$ (for the multiblock case $I_j = \{1, \ldots, m_j\}$) be the index set of the block constraint rows; $\underline{I}_o \subseteq I_o$, $\overline{I}_o = I_o - \underline{I}_o$ be the index sets of the rows of $\underline{D}_o(y)$ and $\overline{D}_o(y)$ respectively and $\underline{I} \subseteq I$, $\overline{I} = I - \underline{I}$ be the index sets of the rows of $(\underline{B}' \quad \underline{D}')$ and $(\overline{B}' \quad \overline{D}')$ respectively.

Using the above notation the *Kuhn–Tucker Stationary Point Problem (KTP)* [7] corresponding to P is defined as:

KTP(P): Find $(x^t, y^t) \in S$ and $(u^t_o, u^t) = (\underline{u}^t_o, \overline{u}^t_o, \underline{u}^t, \overline{u}^t) \in \mathbb{R}^{m_o + m}$, if if they exist, such that

$$\nabla_y f(x^t, y^t) + \underline{D}^t \underline{u}^t + \overline{D}^t \overline{u}^t + \nabla_y D_o(y^t)\, u^t_o = 0 \tag{3.6}$$

$$\nabla_x f(x^t, y^t) + \underline{B}^t \underline{u}^t + \overline{B}^t \overline{u}^t = 0 \tag{3.7}$$

$$\overline{u}^t_o (\overline{D}^t_o(y^t) - \overline{h}^t_o) + \overline{u}^t(\overline{B}^{t\prime} x^t + \overline{D}^{t\prime} y^t - \overline{h}^t) = 0 \tag{3.8}$$

$$(\underline{u}^t_o, \overline{u}^t_o, \underline{u}^t, \overline{u}^t) \geq 0. \tag{3.9}$$

The corresponding problem for PI^t is:

KTP(PI^t): Find $\bar{x}^t \in X\,(y^t)$ and $(\underline{u}^t, \bar{u}^t)$, if they exist, such that

$$\nabla_x f(\bar{x}^t, y^t) + \underline{B}^t \underline{u}^t + \bar{B}^t \bar{u}^t = 0 \tag{3.10}$$

$$\bar{u}^t (\bar{B}^{t\prime} x^t + \bar{D}^{t\prime} y^t - \bar{h}^t) = 0 \tag{3.11}$$

$$(\underline{u}^t, \bar{u}^t) \geq 0. \tag{3.12}$$

Definition 3.2: A point which solves the *KTP* for any given problem P' with nontrivial $(u_0^t,\ u^t)$ will be called a *Constrained Stationary Point (CSP)* of P'. □

In general, we will require that PI^t is solved to obtain x^t as a *SP* or *CSP* of PI^t. The use of either the *Gradient Projection Method (GPM)* (Rosen [8]) or its modification, *Conjugate GPM (CGPM)* (Goldfarb [9]) incorporating Davidon's variable metric approach, is advantageous. Various quantities computed by *GPM* or *CGPM* in the course of solving PI^t will be directly employed in formulae for the proposed method.

Since a *feasible* point $(\bar{x}^t,\ y^t)$ exists, we seek to obtain $(\bar{x}^{t+1},\ y^{t+1})$ such that $f(\bar{x}^{t+1}, y^{t+1}) < f(\bar{x}^t, y^t)$. We define

$$\bar{x}^{t+1} = \bar{x}^t + \bar{v}^t \qquad ; \quad y^{t+1} = y^t + w^t \tag{3.13}$$

where $\bar{v}^t$ and w^t denote the change in x and y respectively. The latter can be normalized in the form $(\bar{v}^t,\ w^t) = \bar{\tau}^t s^t$, where $\bar{\tau}^t = \|(\bar{v}^t,\ w^t)\| > 0$ is the "step length" and $\bar{s}^t = (\bar{v}^t,\ w^t)/\bar{\tau}^t$ is the "direction"; $\bar{s}^t$ is a "feasible direction" (Zoutendijk [10]) if there exists $\bar{\tau}^t > 0$ such that $(\bar{x}^{t+1}, y^{t+1}) = (\bar{x}^t, y^t) + \bar{\tau}^t \bar{s}^t \in S$. A feasible direction is "usable" if in addition $\nabla f(\bar{x}^t,\ y^t) \cdot \bar{s}^t < 0$, so that $f(\bar{x}^{t+1}, y^{t+1}) < f(\bar{x}^t, y^t)$. Thus, in this terminology, we attempt to define a method of selecting a usable direction or step to move from $(\bar{x}^t, y^t)$ to $(\bar{x}^{t+1}, y^{t+1})$.

Such a usable direction, indeed the "best" step $(v^*,\ w^*)$, will be available from P when expressed in terms of $(v,\ w)$. This *Equivalent Problem at* $(\bar{x}^t, y^t)$ is obtained by using (3.13):

EP^t: Find $(v^*,\ w^*)$, if it exists, such that

$$f^*(v^*,\ w^*) = \min\,\{f^t(v,\ w) \mid (\bar{x}^t + v, y^t + w) \in \underline{S}^t = \bigcap_{j=0}^{k} S_j^t\} \tag{3.14}$$

$$\underline{S}_o^t = \{v_j \in \mathbb{R}^{n_j}; j = 1, ..., k;\ y^t + w \in Y\} \tag{3.15}$$

$$\underline{S}_j^t = \{v_j \in \overline{S}_j; y^t + w \in Y \mid B_j' v_j + D_j' w \leq h_j\}; j = 1, ..., k \tag{3.16}$$

$$h_j^{*t} = h_j - B_j' \overline{x}^t - D_j' y^t;\ j = 1, ..., k\ . \tag{3.17}$$

Unfortunately, EP^t is a problem with as many variables as P and would involve as much work as the original problem. The point $(\overline{x}^t, y^t)$ is a CSP of P if $(v^*, w^*) = 0$ in EP^t.

At $(\overline{x}^t, y^t)$ either $x^t \in \text{Int}(X(y^t))$ or $Bdy(X(y^t))$. We will assume that the latter holds and leave the former case for discussion in section 5.

The point $(\overline{x}^t, y^t)$ necessarily lies in the intersection of the q (>0) hyperplanes defined by the solution to PI^t:

$$\underline{B}^{t\prime} x + \underline{D}^{t\prime} y = \underline{h}^t\ . \tag{3.18}$$

In choosing the next point (x^{t+1}, y^{t+1}) we will, quite arbitrarily, require that it satisfies (3.18), i.e. $\underline{B}^{t\prime}(\overline{x}^t + v) + \underline{D}^{t\prime}(y^t + w) = \underline{h}^t$ or equivalently require that:

$$\underline{B}^{t\prime} v + \underline{D}^{t\prime} w = 0\ . \tag{3.19}$$

We will denote this $(n + n_o - q)$-dimensional subspace* of $\mathbb{R}^n$, formed by the intersection of q linearly independent hyperplanes parallel to (3.18), as Q^t (Q_j^t for the multiblock case). The relationship imposed on the choice of (v, w) is simply to require that it lies in Q^t. Thus, we may define a *Modified Equivalent Problem at* (x^t, y^t) by incorporating the restriction (3.19) into EP^t:

MEP^t: Find $(\hat{v}^t, \hat{w}^t)$, if it exists, such that:

$$f^t(\hat{v}^t, \hat{w}^t) = \min\{f^t(v, w) \mid (x^t + v, y^t + w) \in S^t = \bigcap_{j=1}^{k} S_j^t\} \tag{3.20}$$

$$S_o^t = \{v_j \in \mathbb{R}^{n_j};\ j = 1, ..., k;\ y^t + w \in Y\} \tag{3.21}$$

$$S_j^t = \left\{ v_j \in \overline{S}_j;\ y^t + w \in Y \;\middle|\; \begin{matrix} \underline{B}_j^{t\prime} v_j + \underline{D}_j^{t\prime} w = 0 \\ \overline{B}_j^{t\prime} v_j + \overline{D}_j^{t\prime} w \leq \overline{h}_j^{*t} \end{matrix} \right\} \quad ; j = 1, ..., k \tag{3.22}$$

* Q^t designates a subspace in (v, w)-space and a linear manifold in (x, y)-space.

$$\overline{h}_j^{*t} = \overline{h}_j^t - \overline{B}_j^{t\prime}\overline{x}_j^t - \overline{D}_j^{t\prime} y^t \, . \tag{3.23}$$

Two obvious results relate EP^t and MEP^t: (1) (v^t, w^t) solves EP^t if and only if $(\hat{v}^t, \hat{w}^t)$ solves MEP^t and $\underline{u}^t \geq 0$, since *KTP*(MEP^t) is the same as *KTP*(EP^t) except for the nonnegativity restriction on the $\underline{u}$, and (2) $f^t(v^*, w^*) \leq f^t(\hat{v}^t, \hat{w}^t)$, since $S^t \subseteq \underline{S}^t$.

MEP^t is computationally just as unattractive as EP^t or P. Suppose, however, that a solution to MEP^t is attempted using *GPM* or *CGPM*. Any $(\hat{v}^t, \hat{w}^t) \neq 0$ for which $f(\overline{x}^t + \hat{v}^t, y^t + \hat{w}^t) < f(\overline{x}^t, y^t)$ is a candidate for a "usable" direction, for we must have $\nabla f(\overline{x}^t, y^t) \cdot (\hat{v}^t, \hat{w}^t)/\|(\hat{v}^t, \hat{w}^t)\| < 0$. Although in practice one would not have to insist on optimality of MEP^t for the purpose of obtaining an improved point, it will be required in this paper for showing that the solution to P is obtained after a finite number of *MEP* problems. The "best" direction for each manifold Q^t is given by the optimal solution to MEP^t, which theoretically may not be a finite procedure except when f is quadratic and S polyhedral.

Two possibilities may arise:

a) The optimal solution to P, $(x^*, y^*) \in Q^t$, so that $Q^* = Q^t$ is an "optimal face". This is indicated by the nonnegativity of $\underline{u}^*$.

b) $(x^*, y^*) \notin Q^t$ as evidenced by $(\hat{\underline{u}}^{t+1})_i < 0$ for at least one $i \in \underline{I}^t$ so that $Q^t \neq Q^*$. Thus, a new manifold Q^{t+1} is sought such that

$$f(\hat{x}^{t+2}, \hat{y}^{t+2}) \leq f(\hat{x}^{t+1}, \hat{y}^{t+1}); \;\; (\hat{x}^{t+1}, \hat{y}^{t+1}) \in Q^t, \;\; (\hat{x}^{t+2}, \hat{y}^{t+2}) \in Q^{t+1} .$$

We choose $\underline{I}^{t+1} = \underline{I}^t - \underline{I}^{t-}$ where $\underline{I}^{t-} \in \{i | (\hat{\underline{u}}^{t+1})_i < 0\}$. If $f(\hat{x}^{t+1}, \hat{y}^{t+1}) < f(\hat{x}^t, \hat{y}^t)$, we choose $\underline{I}^{t+1} = \{i | \, (B'\hat{x}^{t+1})_i + (D'\hat{y}^{t+1})_i = (h)_i\} - \underline{I}^{t-}$. That is, we replace Q^t by the intersection of all constraints satisfied as equalities at $(\hat{x}^{t+1}, \hat{y}^{t+1})$ except all, or at least one, of those with $(\hat{\underline{u}}^{t+1})_i < 0$.

The rather intuitive approach to improving Q^t from one cycle to the next has been exploited by various methods recently classified as Manifold Optimization or Restriction Methods (Zangwill [11], Geoffrion [1].)

It is stated in the latter that this approach is "principally useful for problems with many nonnegative variables". Thus, it is assumed that the restricted constraints are chosen from the set of orthogonal hyperplanes defining the nonnegativity of the variables. It is precisely this assumption that could be the key to efficiency for *linear* problems. However, it is well known in practice that for *nonlinear* problems the number of active constraints is usually much smaller than the number of variables in the problem. Consequently, partitioning methods using elimination of nonnegative variables based on this assumption are not effective since the reduction of variables, not of constraints, is desirable for structured nonlinear programs.

Thus, the reduction of P into a sequence of *MEP* problems will not be computationally feasible *unless* effective means of drastically reducing the number of variables are devised. In our case such a manipulation is possible and will be based on the linear independence of the rows of $\underline{B}_j^{t'}$ which is identified by *GPM* or *CGPM* as a part of the PI^t solution. The inverse $(\underline{B}_j^{t'}\underline{B}_j^t)^{-1}$ is also readily available. A study of the linear system of constraints in (3.22), in terms of the stated properties of $\underline{B}^{t'}$, leading to a smaller system is presented in *appendix 1.*

In view of these results and the fact that rank $(\underline{B}^{t'}) = q$, we may apply the *Process of Elimination* (*definition A.2*) with $\overline{p} = 0$ to solve for v_j, which limits (v_j, w) to the n_0-dimensional subspaces $\overline{Q}_j^t \subset Q_j^t$, defined by:

$$v_j = -R_j^{t'} w;\ \ j = 1, \ldots, k \tag{3.24}$$

$$R_j^{t'} = (\underline{B}_j^{t'})^\dagger\, \underline{D}_j^{t'} = \underline{B}_j^t\, (\underline{B}_j^{t'}\, \underline{B}_j^t)^{-1}\, \underline{D}_j^{t'};\ \ j = 1, \ldots, k \tag{3.25}$$

and substitute (3.24) into the remaining constraints of MEP^t. This is equivalent to using the projection operator

$$T_j = \begin{pmatrix} 0 & 0 \\ -R_j^t & \mathrm{I} \end{pmatrix} \tag{3.26}$$

to project the normals to the hyperplanes defining S_j^t in (3.22) onto the subspace $v_j = 0$ along the orthogonal complement of the subspace $\overline{Q}_j^t$ as:

$$T_j \begin{pmatrix} \underline{B}_j & \overline{B}_j \\ \underline{D}_j & \overline{D}_j \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & (\overline{D}_j^{t'} - \overline{B}_j^{t'} R_j^{t'})' \end{pmatrix} \tag{3.27}$$

The resulting feasible region defined by:

$$S_j^{pt} = \{y^t + w \in Y \mid (\overline{D}_j^{t'} - \overline{B}_j^{t'} R_j^{t'})w \leq \overline{h}_j^{*t}\};\ j = 1, \ldots, k \tag{3.28}$$

which may be empty if and only if $S_j^t = \emptyset$, represents the orthogonal projection of $S_j^t \cap \bar{Q}_j^t$ onto the subspace defined by $v_j = 0$ (fig. 3.1).

"Backsubstitution" of any point $w^t \in S_j^{pt}$ into (3.24) to obtain the corresponding v_j^t is similarly equivalent to the projection operation

$$T_j' \begin{pmatrix} 0 \\ w^t \end{pmatrix} = \begin{pmatrix} -R_j^{t'} w^t \\ w^t \end{pmatrix} = \begin{pmatrix} v_j^t \\ w^t \end{pmatrix} \tag{3.29}$$

giving a unique $(v_j^t, w^t) \in S_j^t \cap \bar{Q}_j^t$ of minimum Euclidean norm (*theorem A.2*).

In order to express MEP^t completely in terms of w, it is necessary to confine the function and gradient evaluations only to those points satisfying (3.24). This, of course, will be the case for any $y^t + w \in \bigcap_{j=0}^{k} S_j^{pt}$ and the corresponding (unique) v_j given by (3.24). However, in certain instances it is computationally advantageous to introduce this requirement into the definition of the function:

$$F(w) = f^t(\bar{x}^t - R^{t'} w, y^t + w) + f_0^t(y^t + w) \tag{3.30}$$

and use the chain rule to state:

$$\nabla F(w) = -R^t \nabla_v f^t(\bar{x}^t + v, y^t + w) + \nabla_w f(\bar{x}^t + v, y^t + w) \,. \tag{3.31}$$

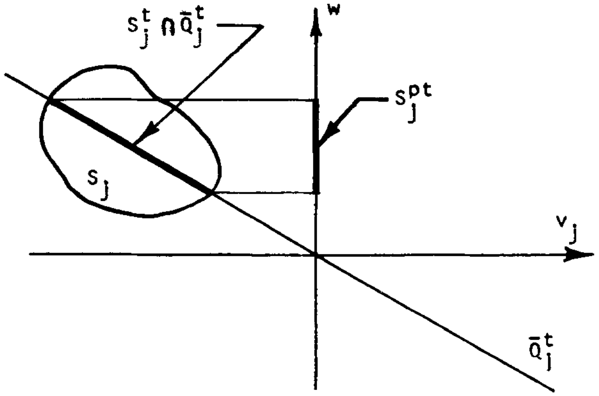

Fig. 3.1. Projection of $S_j^t \cap \bar{Q}_j^t$.

Equivalently, we may project the gradient by

$$T\begin{pmatrix}\nabla_v f\\ \\ \nabla_w f\end{pmatrix} = -R^t \nabla_v f + \nabla_w f . \tag{3.32}$$

In terms of these relations, we define the reduced problem, to be referred to as *Problem II at* $(\bar{x}^t, y^t)$, as:

PII^t: Find (w^t) if it exists, such that

$$F(w^*) = \min \{F(w) \mid w \in S^{pt} = \bigcap_{j=0}^{k} S_j^{pt}\} \tag{3.33}$$

$$S_o^{pt} \equiv S_o^t \tag{3.34}$$

$$S_j^{pt} = \{y^t + w \in Y \mid \bar{H}_j^{t\prime} w \leq \bar{h}_j^{*t}\};\ j = 1, \ldots, k \tag{3.35}$$

$$\bar{H}_j^{t\prime} = (\bar{D}_j^{t\prime} - \bar{B}_j^{t\prime} R_j^{t\prime});\ j = 1, \ldots, k \tag{3.36}$$

$$\bar{h}_j^{*t} = h_j^t - \bar{B}_j^{t\prime} \bar{x}_j^t - \bar{D}_j^{t\prime} y^t;\ j = 1, \ldots, k . \tag{3.37}$$

If P is a convex program, then *PII*^{t} is also a convex program since $F(w)$ has been obtained from $f(x, y)$ by a linear transformation. The solution to *PII*^{t} may be obtained by using any of the available efficient and computationally successful methods. However, since *PII*^{t} generally will have a small number of variables and a rather large number of constraints, computational intuition suggests the use of a method in the dual space such as *GPM* or *CGPM*.

The *KTP* problem corresponding to *PII*^{t} is of interest since it will provide a basis for detecting solutions to P.

KTP(PII^t*)*: Find $w^t \in S^{pt}$ and $(\underline{u}_o^t, \underline{u}_o^t, \bar{u}^t)$ such that

$$\nabla_w F(w^t) + \bar{H}^t \bar{u}^t + \nabla_w D_o(y^t + w^t) \cdot \bar{u}_o^t = 0 \tag{3.38}$$

$$\bar{u}_o^t(\bar{D}_o(y^t + w^t) - \bar{h}_o^t) + \bar{u}^t(\bar{H}^{t\prime} w^t - \bar{h}^{*t}) = 0 \tag{3.39}$$

$$(\underline{u}_o^t, \bar{u}_o^t, \bar{u}^t) \geq 0 . \tag{3.40}$$

Lemma 3.1: Let the current *PII*^{t} be given by (3.33)–(3.37) where $F(w)$ is continuously differentiable and let S^{pt} be bounded. Suppose w^t is a solution to this problem obtained by *GPM* and that $(\underline{u}_o^t, \overline{u}_o^t, \overline{u}^t)$ is the vector of dual variables obtained as part of the solution. Then w^t is a *CSP of PII*^{t}.

Proof: The Lemma follows from the equivalence of the *GPM* termination criteria and *KTP(PII*^{t}*)* (Mangasarian [12], Rosen [4]). □

The partitioning of P into *PI* and *PII* is obviously advantageous since at no time would a problem with as many variables as P have to be explicitly treated. However, this advantage would be of no value unless the solutions to these smaller problems gave means for recognizing the optimality conditions *KTP(P)*. The following generalization of *theorem 2* in [4] illustrates this point.

Theorem 3.1: Let a feasible y^t be given and let $\overline{x}^t$ be a *CSP of PI*^{t}. Then $(\overline{x}^t, y^t)$ is a *CSP of P* if and only if $w^t = 0$ is a *CSP of PII*^{t} and $\underline{u}^t \geq 0$ where

$$\underline{u}^t = (-\underline{B}^{t\prime})^{\dagger\prime}\,(\nabla_x f(\overline{x}^t, y^t) + \overline{B}^t \overline{u}^t)\,. \tag{3.41}$$

Proof: We consider *KTP(P)*, *KTP(PII*^{t}*)*. To show sufficiency, we note that from (3.24) $v^t = 0$ for $w^t = 0$ and thus $(\overline{x}^t + v^t, y^t + w^t) \in S$. Similarly, (3.39) for $w^t = 0$, in conjunction with (3.37) gives (3.8). Premultiplying (3.41) by $(\underline{B}^{t\prime}\underline{B}^t)$ gives

$$\underline{B}^{t\prime}(\underline{B}^t \underline{u}^t + \nabla_x f^t(\overline{x}^t, y^t) + \overline{B}^t \overline{u}^t) = 0 \tag{3.42}$$

for which the second term vanishes since by assumption $\overline{x}^t$ is a *CSP of PI*^{t}. This is precisely (3.7). The conditions (3.6) are shown to result from (3.38) as follows. Applying the chain rule to $F(w)$ gives (3.31) which for $w^t = 0$ becomes:

$$\nabla_w F(w^t) = -R^t\, \nabla_x f(x^t, y^t) + \nabla_y f(x^t, y^t)\,. \tag{3.43}$$

Upon substituting (3.43) into (3.38), using (3.36), (3.25) and (3.41), we obtain (3.6). Finally, (3.9) results from (3.40) and the assumption that x^t is a *CSP of PI*^{t}.

For necessity, observe that $(y^t + w^t) \in Y$ when $w^t = 0$, and that $(x^t, y^t) \in S$ gives (3.24) via *definition A.2, lemma A.1* and thus $w^t = 0 \in S^{pt}$.

Similarly, (3.39) results from (3.8) for $w^t = 0$ by using (3.24), (3.25), (3.36), (3.37). Premultiplying (3.7) by $(\underline{B}^{t\prime}\underline{B}^{t})^{-1}\underline{B}^{t\prime}$ gives (3.41); premultiplying (3.7) by $-R^t$, adding the resulting relation to (3.6) and using (3.24), (3.25) and (3.31) gives (3.38). The nonnegativity of the Lagrange multipliers (3.40) follows from (3.9). □

Corollary 3.1: If $w^t = 0$ is an (unconstrained) stationary point of $F(w)$, then $(\bar{x}^t, y^t)$ is a *CSP of P*.

Proof: Since $w^t = 0 \in \mathrm{Int}(S^{pt})$ solves *KTP(PII^t)*, then $\bar{u}^t = 0$ and, in view of (3.24)–(3.25), $x^{t+1} = \bar{x}^t$ solves *KTP(PI^t)*. Thus,

$$\nabla_x f(\bar{x}^t, y^t) + \underline{B}^t \underline{u}^t + \bar{B}^t \bar{u}^{t\prime} = 0 \tag{3.44}$$

which is precisely (3.7). Also, from (3.43) and assumption:

$$\nabla_w F(w^t) = -R^t \nabla_x f(\bar{x}^t, y^t) + \nabla_y f(\bar{x}^t, y^t) = 0. \tag{3.45}$$

Using (3.25) and (3.44) relation (3.45) gives (3.6). The remaining optimality conditions follow from the proof to *theorem 3.1.* □

Corollary 3.2: Let $f(x, y)$ and $D_0(y)$ be differentiable and convex at $(\bar{x}^t, y^t)$. Then, $(\bar{x}^t, y^t)$ solves P if $w^t = 0$ is an unconstrained minimum of $F(w)$.

Proof: A straightforward application of *KTP(P)* to *corollary 3.1.* □

The above results establish optimality criteria of computational relevance even for nonconvex problems. If $w^t = 0$ and $\underline{u}^t \geq 0$ given by (3.42), the point $(\bar{x}^t, y^t)$ is the desired *CSP* (or global minimum in case f and D_0 are convex).

4. The partitioning algorithm

The algorithm described in this section generally follows Rosen's procedure in [4]. The principal differences lie in the construction and content of *PII* and the special use of *PI* for bringing hyperplanes in and out of the subproblem set of active constraints. However, the underlying process is quite different than Rosen's in that no use of the basis properties of the subproblems is made for eliminating the subproblem variables. Instead, the *Process of Elimination* (*definition A.2* in *appendix 1*), is used in conjunction with the *pseudoinverse* of the matrix of the active constraints for the subproblem. Due to the properties of this operation, it may not generally be assumed that the solution of PII^t and the relations (3.24), (3.13) will necessarily yield a solution to MEP^t. This difficulty is circumvented by requiring that the set of inequalities fixed as equalities in MEP^t remains the same for several additional cycles until the minimum in this fixed subspace, earlier denoted by Q^t, is obtained. We denote this sequence of "subcycles" for a particular "cycle" t by $\{(x^{t_r}, y^{t_r})\}$; $r = 1, 2, \ldots$ In general, $\{(x^{t_r}, y^{t_r})\}$ will be an infinite sequence even when relatively strong differentiablity and convexity assumptions on $f(x, y)$ are made. This point, along with an accelaration procedure which will guarantee finiteness of $\{(x^{t_r}, y^{t_r})\}$ will be discussed in the next section.

We may now outline the iterative solution in terms of a typical partitioning "cycle" t and the "subcycles" t_r. We start a "cycle" with a *feasible* y^t. The subproblem PI^t for this value of y^t is solved to obtain $\bar{x}^t$, the active constraint set $\underline{I}^t$ and the levels of the dual variables $\underline{u}^t$. The corresponding PII^t for this value of $\bar{x}^t$ and for a fixed subspace Q^t defined by the set of indices $\underline{I}^t$, is constructed and solved to obtain the step w^t in the y-space. Using the relations (3.24)–(3.25), which enforce the choice of $\bar{Q}^t \subseteq Q^t$, to compute v we obtain the next point $(x^{t+1}, y^{t+1}) = (\bar{x}^t, y^t) + (v^t, w^t)$ with a step length $\tau^t = \|(v^t, w^t)\|$. A new feasible point is obtained if $\tau^t > 0$ with a reduced function value. It is *feasible*, since PII^t along with (3.24) contains all the constraints of P. If $w^t = 0$ and $\underline{u}^t$ as computed by (3.41) is nonnegative, the optimal solution to P is given by $(\bar{x}^t, y^t)$. Otherwise, we define the next PI for this value of y and repeat the procedure for the next cycle $t + 1$.

It is worthwhile to note again that the solutions to each PI and PII may be an infinite sequence whose convergence to a CSP, and whose convergence under appropriate convexity assumptions to a minimum, is assumed here. Finiteness of this sequence when obtained via $CGPM$ is shown in [9] for quadratic functions.

A summary of the "finite" algorithm follows:

Step 0. Let a *feasible* $y^\circ \in Y$ be given. Let $y^t = y^\circ$, $E^t = \emptyset$, $t = 0$, $r = 0$ where E^t denotes the set of constraint indices defining Q^t.

Step 1. Define and solve the subproblem $PI^{t_r} = PI(E^t, y^{t_r})$. This gives the solution $\bar{x}^{t_r}$. If $E^t = \emptyset$, let $E^t = \underline{I}^t$, compute R^t by (3.25), go to step 3. Otherwise, go to step 2.

Step 2. (2a): If $c^{t_r} = (\bar{x}^{t_r} - x^{t_r}) = 0$, then the point (x^{t_r}, y^{t_r}) solves MEP^{t-1}. Set $E^t \neq \emptyset$, $r = 0$, go to step 1. (2b): If $c^{t_r} \neq \emptyset$, set $\bar{x}^{t_r} \rightarrow x^{t_r}$ go to step 3.

Step 3. Define and solve PII^{t_r} at $(\bar{x}^{t_r}, y^{t_r})$. This gives w^{t_r} and the function value $F(w^{t_r})$. Compute $\underline{u}^{t_r}$ via (3.41). If $w^{t_r} = 0$ and $\underline{u}^{t_r} \geq 0$, then the point (x^{t_r}, y^{t_r}) solves P. Stop. If $w^{t_r} \neq 0$, *or* $w^{t_r} = 0$ *and* there exists at least one i such that $(\underline{u}^{t_r})_i < 0$, then the next point is obtained as follows:

If this step was entered from step 2b, compute:

$$v^{t_r} = -R^{t'} w^{t_r} \tag{4.1}$$

$$x^{t_r+1} = x^{t_r} + v^{t_r} \; ; \quad y^{t_r+1} = y^{t_r} + w^{t_r} \tag{4.2}$$

so that

$$f(x^{t_r+1}, y^{t_r+1}) = F(w^{t_r}) \leq f(x^{t_r}, y^{t_r}) \, . \tag{4.3}$$

Let $r + 1 \rightarrow r$ and return to step 1 to start another "subcycle". If this step was entered from step 1, compute:

$$v^{t_r} = -R^{t'} w^{t_r} \tag{4.4}$$

$$x^{t+1} = \bar{x}^{t_r} + v^{t_r} \; ; \quad y^{t+1} = y^{t_r} + w^{t_r} \tag{4.5}$$

so that

$$f(x^{t+1}, y^{t+1}) = F(w^{t_r}) \leq f(x^t, y^t) \tag{4.6}$$

Let $E^{t+1} = E^t$, $t + 1 \rightarrow t$ and start another "cycle" at step 1. □

We note that effectively, the determination of the step at (x^{t_r}, y^{t_r}) for the t_r-th "subcycle" is made in two stages. First, by solving PII^{t_r-1} and using (3.24), we obtain the step (v^{t_r-1}, w^{t_r-1}) and the corresponding point (x^{t_r}, y^{t_r}) via (3.13). Then, by solving PI^{t_r} we obtain a step c^{t_r} in the x-space which modifies (v^{t_r-1}, w^{t_r-1}) to $(\bar{v}^{t_r-1}, w^{t_r-1}) = (v^{t_r-1} + c^{t_r}, w^{t_r})$ resulting in the point $(\bar{x}^{t_r}, y^{t_r})$. This is depicted in fig. 4.1. The same procedure is applicable for the determination of the step for a "cycle". Clearly if $\|w^{t_r}\| = \epsilon > 0$, then $\|(\bar{v}^{t_r}, w^{t_r})\| \geq \epsilon$.

Lemma 4.1: Let $f(x, y)$ and $D_0(y)$ be continuously differentiable and let a constraint qualification be satisfied for Y. Then,

$$f(x^{t+1}, y^{t+1}) \leq f(x^t, y^t) \tag{4.7}$$

where, (x^t, y^t) and (x^{t+1}, y^{t+1}) denote the values of (x, y) at the t-th and $(; + 1)$-th "cycles" respectively.

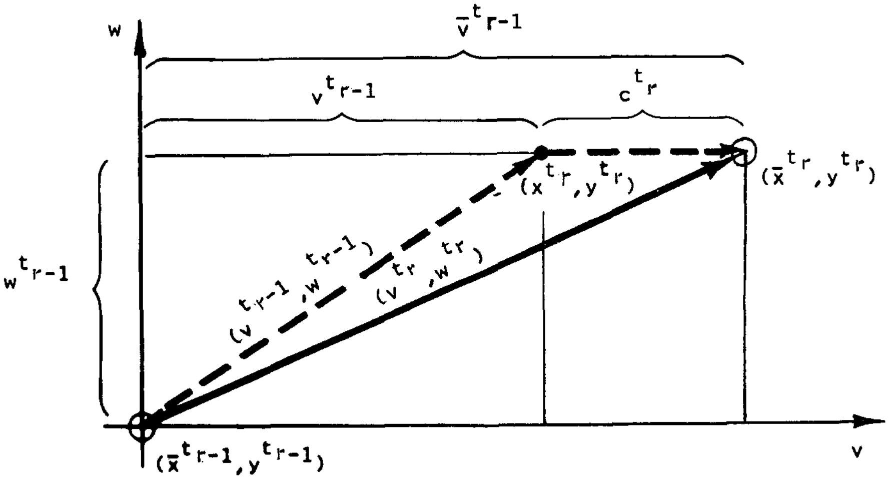

Fig. 4.1. Effective subcycle step.

If the normals to the hyperplanes active at (x^t, y^t) are linearly independent, strict inequality holds in (4.7).

Proof: In the $(t-1)$-th "cycle", we obtained an optimal solution to MEP^{t-1} so that $(x^t, y^t) \in Q^{t-1} \subseteq S^{t-1}$ with $f(x^t, y^t)$. Subsequently, we let $E^{t-1} = \emptyset$ and sought a minimum to the problem $PI^t \equiv PI(y^t)$ establishing $\bar{x}^t$, and the new $Q^t \Leftrightarrow \underline{I}^t$. Then, fixing Q^t we obtained a minimum $(x^{t+1}, y^{t+1}) \in Q^t$, after several "subcycles", with $f(x^{t+1}, y^{t+1})$.

Clearly then, each "cycle" either lowers or leaves unchanged the value of $f(x, y)$. Hence, $\{f(x^t, y^t)\}$ is nonincreasing.

If $\underline{u}^t \geq 0$, then (x^t, y^t) solves P. If there are no redundant active constraints at (x^t, y^t) for which $(\underline{u}^t)_i < 0$ for at least one $i \in \underline{I}^{t-1}$ then PI^t will result in the new set of active constraints $\underline{I}^t \neq \underline{I}^{t-1}$ (and $Q^t \neq Q^{t-1}$). Subsequent minimization in Q^t will provide (x^{t+1}, y^{t+1}) such that $f(x^{t+1}, y^{t+1}) < f(x^t, y^t)$. □

Theorem 4.1: Let $f(x, y)$ and $D_0(y)$ be differentiable and convex and let a constraint qualification be satisfied for the set Y. Furthermore, let the normals to the hyperplanes active at each "cycle" be linearly independent. Then, if P has an optimal solution it is obtained in a finite number of "cycles".

Proof: First note that if the algorithm terminates, the point (x^t, y^t) must be an optimal solution to P, for if not the nonnegativity test on $\underline{u}^t$ would fail, i.e. the *KTP(P)* would not have a solution at (x^t, y^t, u^t), initiating the next "cycle".

To show finiteness of "cycles", we note that, by assumption and *lemma 4.1*, $f(x^{t+1}, y^{t+1}) < f(x^t, y^t)$ for $(x^t, y^t) \in Q^{t-1}$ and $(x^{t+1}, y^{t+1}) \in Q^t$ and note that $Q^{t-1} \neq Q^t$. It is clear that the proof of *lemma 4.1* insures the impossibility of returning to the same Q at any later cycle s with a lower function value, i.e. $Q^s \neq Q^{t-1}$ for all $s > t-1$. There are, of course, only a finite number of possible choices for Q^s since P has a finite number of constraints. If follows that after a finite number of cycles an optimal solution to P is obtained. □

Remark 4.1: The proof of *lemma 4.1* shows that the manifold Q is altered in two steps. First, for y fixed, by canceling the requirement that all constraints with indices $i \in E^{t-1}$ are kept as equalities, the larger feasible domain $X(y^t)$ is obtained. A minimization over this domain gives a point $(\bar{x}^t, y^t)$ such that $f(\bar{x}^t, y^t) < f(x^t, y^t)$, except when there are redundant constraints at $(\bar{x}^t, y^t) \in X(y^t)$. In this case, a possibility of cycling exists. Nevertheless, it can easily be prevented by applying one of the well known perturbation techniques (see e.g. Dantzig [13]). Subsequent "subcycle" minimizations in *PI* and *PII* can only improve or leave the function value unchanged. Therefore, in practice, a strict inequality in (4.7) may be assumed.

Remark 4.2: In section 2 boundedness of the feasible region S was assumed. In the absence of this assumption, the above algorithm gives a practical procedure for detecting an unbounded solution to P. If P has an unbounded solution, from the proof of theorem 4.1 it follows that a problem MEP^t is encountered after a finite number of cycles with an unbounded solution.

5. Minimization in the linear manifold Q^t

The special case of a problem P with a stationary or minimum point (x^*, y^*) such that $x^* \in X(y^*)$, is of particular interest since the algorithm reduces to sequential changes in x and y. This is comparable to the simplest scheme ("*sectioning*" or "*one-at-a-time*": D'Esopo [14], Wilde [15]) for minimizing an unconstrained function of several variables by a sequence of minimizations along each variable while keeping all remaining variables constant. Our case is a generalization of this scheme and resembles the method in Warga [16] which uses minimizations with respect to a *subset* of variables while keeping all remaining variables fixed. To be more precise, given a point $(\bar{x}^t, y^t) \in \mathbb{R}^n$ we obtain $\bar{x}^{t+1}$ as: $f(\bar{x}^{t+1}, y^t) = \min_x f(x, y^t)$, and y^{t+1} as: $f(\bar{x}^{t+1}, y^{t+1}) = \min_y f(\bar{x}^{t+1}, y)$. The process is then repeated to obtain $\bar{x}^{t+2}$ with y^{t+1} fixed, etc. Its convergence to a minimum is established in the following

Theorem 5.1 [16]: Let $f(x, y)$ be continuously differentiable and let $f(x, \bar{y})$, $f(\bar{x}, y)$ be strictly convex for any fixed $\bar{y} \in \mathbb{R}^{n_o}$ and $\bar{x} \in \mathbb{R}^n$, respectively. Then, if (x^*, y^*) is a limit point of $\{(x^t, y^t)\}$; $t = 0, 1, 2, \ldots$ $f(x^*, y^*) = \min \{f(x, y) \mid (x, y) \in \bar{S}\}$. □

However, the convergence of the sectioning (one variable at a time) process is known to be very poor and may be intolerable if (x^*, y^*) lies on a "ridge" (see fig. 5.1). Similar, but relatively improved, behavior could be expected in sectioning with a subset of variables.

For a given practical problem P, it is unlikely that (x^*, y^*) will be interior to all block constraints. Nevertheless, the above discussion is relevant in another way. Suppose that a PII^t is solved to give w^t and the corresponding point (x^{t+1}, y^{t+1}) is obtained by (3.24) and (3.13). In general, because of our decision to fix $p = 0$ in (A.1), we may not assume that (x^{t+1}, y^{t+1}) solves MEP^t. As mentioned earlier, in order to obtain a *CSP* or minimum in the manifold Q', several cycles, each requiring separate minimizations in x and y may be necessary. This situation may arise when the optimal face Q^* is reached and $(x^*, y^*) \in Q^*$ is sought.

To improve convergence one must provide certain nonorthogonal directions to modify the d_i in fig. 5.1. The choice of such directions is rather heuristic although several problems have benefited from such

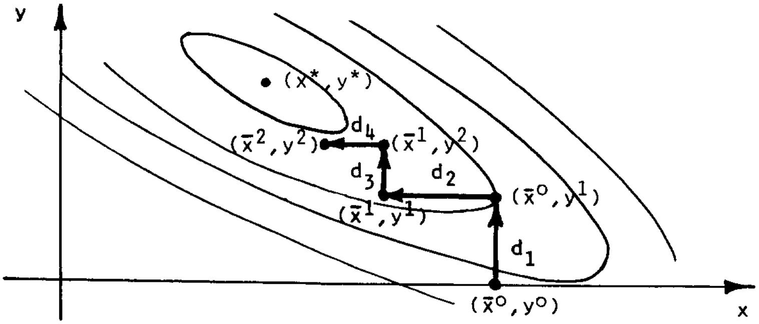

Fig. 5.1. Sectional search.

treatment. For example, in Hooke–Jeeves [17] a *"pattern search"* technique is devised to remain on the "crest of a ridge" and similar "diagonalization" devices are used in Faure–Huard [18].

As an extension of these techniques to our algorithm, we will attempt to enrich the mix of directions available to *PII* by providing some directions based on performance of the previous steps in *PI* (Rosen [19]).

Suppose that the (n, r)-matrix C consisting of linearly independent columns $c^i \in Q^t \subseteq \mathbb{R}^n$; $i = 1, \ldots, r \leq n-q$ is available. In view of *theorem A.1,* we may rewrite (3.24) as:

$$v = -R^{t\prime} w + C\alpha \tag{5.4}$$

where $\alpha \in \mathbb{R}^r$ are newly appended variables. Use of this relationship to derive *PII*, instead of (3.24) results in the *Augmented Problem II (APII) at the point* $(\bar{x}^t, y^t)$:

APII^t: Find $(w^t, \alpha^t) \in \mathbb{R}^{n_o+r}$, if it exists, such that

$$F(w^t, \alpha^t) = \min \{F(w, \alpha) \mid (w, \alpha) \in S^{pt} = \bigcap_{j=0}^{k} S_j^{pt}\} \tag{5.5}$$

$$S_o^{pt} \equiv S_o^t \tag{5.6}$$

$$S_j^{pt} = \{y^t + w \in Y \mid \bar{H}_j^{t\prime} w + \bar{B}_j^{t\prime} c_j \alpha_j \leq \bar{h}_j^{*t}\};\ j = 1, \ldots, k \tag{5.7}$$

$$\bar{H}_j^{t\prime} = (\bar{D}_j^{t\prime} - \bar{B}_j^{t\prime} R_j^{t\prime});\ j = 1, \ldots, k \tag{5.8}$$

$$\bar{h}_j^{*t} = \bar{h}_j^t - \bar{B}_j^{t\prime} \bar{x}_j^t - \bar{D}_j^{t\prime} y;\ j = 1, \ldots, k\ . \tag{5.9}$$

Since $c^i \in Q^t$, for any $(w^t, \alpha^t) \in S^{pt}$ and for the corresponding v^t computed by (5.4), $(v^t, y^t) \in Q^t$ and consequently by (3.13) we obtain $(x^{t+1}, y^{t+1}) \in Q^t$.

The intuitive notion that one needs to incorporate only *some* well chosen direction, leads us to append these extra variables to *PII* one at a time but only whenever Q^t remains the same for several cycles. These variables may be omitted from *PII* whenever their levels remain insignificant. This process, which creates the matrix C in (5.4), is outlined below.

Definition 5.1: The *Acceleration Procedure (AP)* is the following modification of the algorithm of section 4:

Replace step (2b) of the algorithm by:
''2b) If $c^{t_r} \neq 0$, set $\overline{x}^{t_r} \to x^{t_r}$ and let

$$v^{t_r} = -R^{t\prime} w + \sum_{i=1}^{r} c^{t_i} \alpha_i \tag{5.10}$$

Append the variable α_r and column $\overline{B}^{t\prime} c^{t_r}$ to *APII.* ''
The remaining steps of the algorithm are the same except that we read $APII^{t_r}$ for PII^{t_r} and $F(w, \alpha)$ for $F(w)$ whenever $r > 0$. Similarly (4.1) and (4.4) are replaced by (5.10). □

The crucial requirement of linear independence among the columns of C in (5.4) is given in the following

Theorem 5.2: *AP* produces mutually linearly independent directions c^{t_r} for $r < n-q$.

Proof: We have $Q^t \subseteq \mathrm{R}^{n+n_o-q}$ given by the q active constraints (3.19). From the results of *appendix 1*, in particular *lemmas A.1* and *A.2*, we know that the relation (3.24):

$$v = -R^{t\prime} w \tag{5.11}$$

defines a linear subspace $\overline{Q}^t$ such that $\overline{Q}^t \subseteq Q^t$.

The proof will be by induction. Let $c^{t_1} \neq 0$, as defined in *AP*, which has resulted from the PI^{t_1} for a fixed y^{t_1} and consequently corresponds to $w = 0$. Thus, the direction $(c^{t_1}, 0) \notin \overline{Q}^t$ since it does not satisfy (5.10) for $c^{t_1} \neq 0$. Therefore, $(c^{t_1}, 0) \in Q^t$ is linearly independent from all directions in $\overline{Q}^t$. The new PII^{t_1} is then based on

$$v = -R^{t\prime} w + c^{t_1} \alpha_1 \tag{5.12}$$

instead of (5.11).

Now let PII^{t_r}, $r < n-q-1$ be constructed with

$$v = -R^{t\prime} w + \sum_{i=1}^{r} c^{t_i} \alpha_i \tag{5.13}$$

instead of (5.11). Its solution and (5.13) give (v^{t_r}, w^{t_r}) which, via (3.13) gives:

$$x^{t_{r+1}} = x^{t_r} + v^{t_r} \; ; \quad y^{t_{r+1}} = y^{t_r} + w^{t_r} \; . \tag{5.14}$$

We may regard the overall problem in x with fixed y as:

$$f(x^{t_{r+1}}, y^{t_{r+1}}) = \min_{\alpha_i} f(x^{t_r} + v, y^{t_{r+1}}) \tag{5.15}$$

for if $c^{t_{r+1}} = (\bar{x}^{t_{r+1}} - x^{t_{r+1}}) \neq 0$, then the corresponding $w = 0$. It follows that if the $PI^{t_{r+1}}$ solution $(\bar{x}^{t_{r+1}}, y^{t_{r+1}}) \in Q^t$, then $(c^{t_{r+1}}, 0)$ is linearly independent of the space spanned by $c^{t_1}, \ldots, c^{t_r}$ for if not, a different choice of the α_i could have provided a lower function value in PII^t. □

6. Computational aspects

The simplified computational algorithm for the multiblock case is based on the algorithm of the previous section as outlined below.

Step 0. Obtain a *feasible* y^{o} by some initialization procedure. Let $j = 1$ and $t = 0$.

Step 1. Define and solve PI_j^t $(\equiv PI_j(y^t))$. This gives $\bar{x}_j^t$, $\bar{\underline{u}}_j^t$ and the set of active constraints $\underline{I}_j^t$.

Step 2. If $c_j^t = \bar{x}_j^t - x_j^t \neq 0$, compute $\bar{h}_j^{*t}$ by (3.37). (2a): If $\underline{I}_j^t = \underline{I}_j^{t-1}$ let $R_j^t = R_j^{t-1}$, $\bar{H}_j^t = \bar{H}_j^{t-1}$. (2b): If $\underline{I}_j^t \neq \underline{I}_j^{t-1}$ compute R_j^t by (3.25) and $\bar{H}_j^t$ by (3.36).

If $c_j^t = 0$, the j-th block is "optimal". In either case, if $j < k$, set $j + 1 \to j$ and go to step 1. Otherwise, go to step 3.

Step 3. Define and solve PII^t at the point $(\bar{x}_1^t, \ldots, \bar{x}_k^t, y^t)$ to obtain w^t and the function value $F(w^t)$. Compute $\underline{u}_j^t$; $j = 1, \ldots, k$ via (3.43) (see discussion below).

If $w^t = 0$ and $\underline{u}^t \geq 0$; $j = 1, \ldots, k$ then the point $(\bar{x}_1^t, \ldots, \bar{x}_k^t, y^t)$ solves P. Terminate.

If $w^t \neq 0$, the next point is obtained by computing

$$v_j^t = -R_j^{t\prime} w^t; \; j = 1, \ldots, k \tag{6.1}$$

$$x_j^{t+1} = \bar{x}_j^t + v_j^t \; ; \; j = 1, \ldots, k; \; y^{t+1} = y^t + w^t \tag{6.2}$$

so that

$$f(x_1^{t+1}, \ldots, x_k^{t+1}, y^{t+1}) = F(w^t) \leq f(x_1^t, \ldots, x_k^t, y^t). \tag{6.3}$$

Set $j = 1$, $t + 1 \to t$ and go to step 1. □

We note that the computation for $\underline{u}_j^t$ (in step 3) for every cycle is not necessary. For $w^t = 0$, $\underline{u}_j^t = \overline{\underline{u}}_j^t$ which is available from the solution of $PI_j^t; j = 1, ..., k$ in step 1. For $w^t \neq 0$, the level of $\underline{u}_j^t$ is not required since the removal or addition of hyperplanes to the manifold Q^t is handled by solving $PI_j^{t+1}; j = 1, ..., k$.

The *AP* (*Definition 5.1*) amounts to replacing step (2a) above, whenever $c_j^t \neq 0$, by:

"2a) If $\underline{I}_j^t = \underline{I}_j^{t-1}$ let

$$R_j^{t\prime} = (R_j^{(t-1)\prime}, c_j^t) \tag{6.4}$$

$$\overline{H}_j^{t\prime} = (\overline{H}_j^{(t-1)\prime}, \overline{B}_j^{t\prime} c_j^t) \tag{6.7}$$

and append the variable α_j^t to *APII*, i.e. we set $(w, \alpha_j^t) \to w$. "

The remaining parts of the algorithm remain the same except that we read *APII*^{t} for *PII*^{t}.

Formally, the above summary of the algorithm differs from the exposition in section 4. No "subcycles" have been introduced to insure that the minimum in the manifold Q^t is obtained at each "cycle". The exact minimization in Q^t, and thus the solution of *MEP*^{t} was necessary to insure finiteness of the sequence of cycles. Computationally, however, this stringent requirement may be relaxed until the last stages of the optimization process. The resulting effect is intuitively pleasing since otherwise much computational effort would be expended for finding the minimum in a manifold which is not contained in the optimal manifold Q^*. The theoretical implications of this computationally advantageous modification are well known. A possibility exists that, after a number of cycles, one could return to the set of active constraints of an earlier cycle with a lower function value. Thus, the sequence (x^t, y^t) may be infinite.

Various commonly known methods to prevent this behavior may be employed. One such unappealing device consists of recording $\underline{I}^t$ for each cycle t and insisting on minimization in Q^p whenever $\underline{I}^s = \underline{I}^p$; $s < p$. For most problems, this will be unnecessary since the same set of constraints does not normally reappear after a number of cycles except when it is the optimal set. In the latter case, application of *AP* will obtain the optimal point in a finite number of additional cycles.

An experimental computer program * for solving P has been written in *FORTRAN* under the time sharing system *CP/67-CMS* and has been

* This experimental program is for IBM internal use only.

tested on a number of test problems. Although the program is completely core resident, it makes extensive use of peripheral direct access devices for storing the given problem data and intermediate results which are needed in subsequent cycles. Thus, it provides a vehicle for future large scale testing of the partitioning algorithm and various modifications and extensions which may be proposed later.

The program is divided into 25 subroutines which essentially perform the following functions:

(a) The *program and file initialization phase* reads in the problem sizes and performs certain system functions for creating four direct access files A, B, C and D. File A will contain the initial problem data arranged by blocks in the sequence $j = 1, ..., k, 0$. Files B and C will contain intermediate results needed in the following cycle such as: $\bar{x}_j^t$, $\underline{I}_j^t$, classification of subproblem row indices not in $\underline{I}_j^t$ into linearly dependent actives, inactives, etc., $(\underline{B}^{t\prime}\underline{B}^t)_j^{-1}$ (although this is not necessary) R_j^t, $\bar{H}_j^t$, $\bar{h}_j^{*t}$ and some other index arrays. These are arranged in increasing order of $j = 1, ..., k$. In each cycle files A and B will be used alternately to write on one and read from the other. Their roles are switched after the completion of each cycle. File D contains intermediate data from PII^t solution such as selected index arrays, w^t, etc.

(b) The *input phase* for the j-th subproblem reads in the index set for equality constraints (if any), B_j, h_j, upper and lower bounds (if any) on the variables x_j, and D_j. Then, it computes the scaling divisors and normalizes B_j, h_j, D_j. The input of the given function arrays is performed next by calling a portion of a (user written) subroutine defining the function and gradient computations. All of these data are then written on file A. This phase is used only in the first cycle. All subsequent cycles use the setup phase (c).

(c) The *setup phase* for PI_j^t is called to restore the PI_j^{t-1} data from file A into core. Then, it computes the right hand side of the PI_j constraints for the fixed value of $y = y^t$. Optionally, the initial point x^t or $\bar{x}^{t-1}$ and the inverse $(\underline{B}^{(t-1)\prime}\underline{B}^{(t-1)})^{-1}$ of PI^{t-1} are restored.

(d) The *solution of PI_j* is then obtained by calling *CGPM* which in turn uses the function and gradient evaluation routine.

(e) The *PII definition phase* extracts the solution vectors, index arrays, matrices etc. from *CGPM*. It computes R_j^t, $\bar{H}_j^t$, $\bar{h}_j^{*t}$ and stores all the data on file B.

(h) The *PII setup phase* concatenates matrices $\bar{D}_0'$, $\bar{H}_1'$, ..., $\bar{H}_k'$ and right hand sides $\bar{h}_0$, $\bar{h}_1^*$, ..., $\bar{h}_k^*$ to arrange them in a single constraint matrix and right hand side vector for use by *CGPM*. Certain rows of this matrix with zero coefficients and positive right hand side elements are cancelled. Superfluous constraint detection mechanisms may be included at this point to reduce the row size of the matrix further. The function and gradient subroutine is called next to create and store in core the appropriate arrays for the *PII* function $F(w)$.

(i) The *PII solution phase* simply calls *CGPM* which in turn uses the portion of the function and gradient evaluation subroutine pertaining to $F(w)$.

(j) The *"next point" phase* extracts the solution w^t and other relevant index arrays from *CGPM* and computes the new feasible point $(x_1^{t+1}, ..., x_k^{t+1}, y^{t+1})$ by applying (3.24), (3.13) which require a sequential review of x_j^t and R_j^t; $j = 1, ..., k$ on file B.

(k) The *output phase* prepares the final solution report as well as intermediate output, both on the offline printer and the user terminal where the solution path may be monitored.

A number of strictly convex quadratic programming test problems were solved successfully by the experimental program outlined in the preceding section. The data for these problems were randomly generated as in Rosen–Suzuki [20], Grigoriadis [6]. No claims will be made on the resemblance of these to actual industrial problems. In fact, our test cases are too small to allow meaningful inference for problems of much larger sizes for which the proposed partitioning method is primarily intended. The results for the eight test cases presented in table 6.1 should be regarded only as preliminary since various refinements to the computer program, such as strategies for avoiding the solution of all *PI* at each cycle, the *acceleration procedure (AP)*, etc., have not been fully tested and were inoperative during the solution of these test cases. All problems were started with the same "feasible" y^o vector. The initial points $(\bar{x}^o, y^o)$ constructed by the partitioning algorithm and *CGPM* were different in their choice of $\bar{x}^o$. *CGPM*, being a "second order" method, is particularly suited for solving strictly convex quadratic programming problems which accounts for the relatively small number of iterations and function evaluations. Nevertheless, strict comparison of the number of iterations could be misleading since the problem dimensionality for the complete *P* and the *PI* and *PII* problems are quite different. For instance, problem No. 7 requires 73 function evaluations at points $(x, y) \in \mathbb{R}^{36}$, or equivalently 95,000 multiplications as opposed to 487 function evaluations at points $x \in \mathbb{R}^8$, and 66 evaluations for points $y \in \mathbb{R}^4$, or equivalently 40,000 multiplications. Additional

Table 6.1.
Preliminary test results.

	Problem structure					Complete problem P (Total size)		⟨Solution by the proposed partitioning method⟩					Solution by CGPM (Goldfarb [9])	
Test case No.	n_o Coupl. var's	m_o Constr's on y var's only	m_j Constr's in block j	n_j var's in block j	k No. of blocks	m^* – Constr's	n^* – var's	No. of cycles to opt.	Tot. no. of CGPM iter's in *PI*	Tot. no. of *PI* funct. eval's	Tot. no. of CGPM iter's in *PII*	Tot. no. of *PII* funct. eval's	CGPM iter's on complete P.	Funct. eval's for complete P
1	2	6	11	4	2	28	10	3	11	24	5	13	6	10
2	2	6	11	4	4	50	18	6	29	70	7	17	11	14
3	2	6	11	4	6	72	26	2	30	54	1	4	14	18
4	2	6	11	4	8	94	34	3	33	69	6	15	15	18
5	2	6	11	4	10	116	42	5	64	147	5	15	23	26
6	4	12	22	8	2	56	20	6	45	71	8	19	16	21
7	4	12	22	8	4	100	36	23	286	487	27	66	57	73
8	4	12	22	8	5	122	44	9	150	282	16	31	24	28

Table 6.2.
Solution history for problem 7.

t	$\lVert w^t \rVert$	No. blks $w/$ new basis	No. optim. blocks	$\langle$Function val.$\rangle$
0	0.457 30	4	0	−278.371 244
1	0.103 46	3	0	−278.782 080
2	0.016 04	0	0	−278.807 451
3	0.002 40	0	0	−278.810 390
4	0.000 67	0	3	−278.811 177
5	0.143 85	1	2	−278.894 334
6	0.018 16	1	0	−278.906 247
7	0.040 94	1	0	−278.907 790
8	0.028 36	1	0	−278.909 139
9	0.023 42	1	0	−278.909 933
10	0.030 46	0	0	−278.911 071
11	0.039 51	0	0	−278.912 986
12	0.050 88	0	0	−278.916 183
13	0.065 99	0	0	−278.921 556
14	0.034 54	1	0	−278.928 771
15	0.096 35	0	0	−278.936 310
16	0.125 33	0	0	−278.955 610
17	0.023 49	0	0	−278.974 745
18	0.047 45	1	0	−278.980 563
19	0.019 92	0	0	−278.983 745
20	0.008 24	0	0	−278.984 297
21	0.003 65	0	1	−278.984 405
22	0.001 36	0	3	−278.984 422
23	0.000 38	0	4	−278.984 425

effort is, of course, necessary for constructing *PII* whenever the set of active constraints is altered, which in this case was infrequent.

The results appear quite promising since they clearly indicate that implementation of the above refinements, in particular *AP*, will considerably improve convergence. This assumption is based on the experimental observation that the set of active constraints is infrequently altered and that the optimal set is normally identified during the early stages of the optimization procedure. The majority of cycles are performed to locate improved points in the intersection of the active constraints. This behavior was to be expected (see Section 4) since the *AP* was not employed. Table 6.2 illustrates this point for problem 7.

Experimentation with larger convex and nonconvex programming problems via a more sophisticated revision of the present computer program and a substantial amount of computer time will be required to conclusively show the efficiency or superiority of this method over other methods for ordinary nonlinear programs. However, the advantage of the partitioning method lies in its ability to handle large nonlinear problems, perhaps by an order of magnitude, than those handled by any currently available nonlinear programming code. This is due to the fact that a prohibitive amount of computer storage is required for treating such problems directly.

Acknowledgements

The valuable comments offered by Professors J.B. Rosen, O.L. Mangasarian and K. Ritter are greatly appreciated. Helpful discussions were held with Professor D. Goldfarb in conjunction with his CGPM computer program.

Appendix 1

Linear Systems of Equations, the Pseudoinverse and the Process of Elimination

We consider the linear system of equations $Ax = b$, where A is a nonzero real (m, n)-matrix $(m \leq n)$ and b is an m-vector. The system $Ax = b$ is consistent if b is in the column space of A. The notation used in this appendix is independent of that in other sections.

The general solution to this system has been characterized by a number of results originally due to E.H. Moore [22]. It has also been discussed in detail in Penrose [23], Greville [24], Zadeh-Desoer [25]. The pivotal result in this theory is the generalization of the notion of the *inverse* transformation A^{-1} (of the nonsingular linear transformation A: $\mathbb{R}^n \to \mathbb{R}^n$) to the *pseudoinverse* (also called generalized inverse or generalized reciprocal) transformation $A^{\dagger}$.

Definition A.1: Let rank $(A) = m$. The (right) *pseudoinverse* of A is defined as the (n, m)-matrix $A^{\dagger} = A'(AA')^{-1}$. □

It may easily be shown that a unique $A^{\dagger}$ exists.

Theorem A.1: Let rank $(A) = m$ and let p be any vector in $\mathbb{R}^n$. The *general solution* to $Ax = b$, if it exists, is given by:

$$x = A^\dagger b + (\mathrm{I} - A^\dagger A)p \tag{A.1}$$

Proof: First, verify that $x = A^\dagger b$ is a particular solution to $Ax = b$. Second, if $x = (\mathrm{I} - A^\dagger A)p$ for any, $p \in \mathbb{R}^n$, then $Ax = A(\mathrm{I} - A^\dagger A)p = (A - AA^\dagger A)p = 0$ and conversely if $Ax = 0$ then $A^\dagger Ax = x - x$ or $x = (\mathrm{I} - A^\dagger A)x$. The sum of the particular and homogeneous solutions gives (A.1). □

Theorem A.2: Let rank$(A) = m$. Let the general solution to $Ax = b$ exist and be given by $x_o = A^\dagger b + (\mathrm{I} - A^\dagger A)p$ for any $p \in \mathbb{R}^n$. Then, for a fixed $p = \bar{p}$ the solution x_o is unique.

Proof (Zadeh–Desoer [25]): Let $x = x_o + x_1$ for any $x_1 \neq 0$ such that $x_1 \in N(A) = \{x | Ax = 0\}$. Thus $Ax = Ax_o = A(x_o + x_1)$. Let $\bar{k} = (\mathrm{I} - A^\dagger A)\bar{p}$. Then, by definition $x_o \in \mathrm{R}(A^\dagger) = \{(x - \bar{k}) \in \mathbb{R}^n | x - \bar{k} = A^\dagger b$ for all *feasible b*$\}$ and since $R(A^\dagger) = R(A') = N(A)^\perp$ (where $N(A)^\perp$ denotes the linear manifold orthogonal to $N(A)$), we have $x_o \in N(A)^\perp$. Therefore, $\|x\|^2 + \|x_1\|^2 = \|x_o + x_1\|^2$ and since $x_1 \neq 0$, we have $\|x_o\| < \|x\|$. Thus x_o is the only solution which possesses minimum norm. □

The above theorem establishes an important property of the linear mapping $A^\dagger : \mathbb{R}^m \to \mathbb{R}^n$. Due to the fact that rank $(A^\dagger) = m$ $(\leq n)$ and dim $R(A^\dagger) = m$, there exist points $x \in \mathbb{R}^n$ such that no $b \in \mathbb{R}^m$ can be found such that $x = A^\dagger b$. Consequently, $A^\dagger$ may *not* be regarded as a one-to-one mapping. This predictable observation is nevertheless extremely important. Fixing $p = \bar{p}$, is in effect equivalent to artificially introducing $(n-m)$ constraints so that $R(A^\dagger)$ does not span the entire $\mathbb{R}^n$. The only way that *any* $x \in \mathbb{R}^n$ may be expressed in terms of $A^\dagger$ is by introducing $p \in \mathbb{R}^n$ as variables in the general solution to $Ax = b$. Since $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$ and $p \in \mathbb{R}^n$, it is necessary to consider only $(n-m)$ components of p in the general solution. In the following we will fix $p = \bar{p}$ and later let $\bar{p} = 0$.

Definition A.2: Let the following consistent system of linear equations be given:

$$A_1 x + A_2 y = b_1 \tag{A.2}$$

$$A_3 x + A_4 y = b_2 \tag{A.3}$$

where A_1, A_2, A_3, A_4 are (m_1, n), (m_1, n_o), (m_2, n) and (m_2, n_o)-submatrices $(m_1 \leq n)$, b_1 and b_2 are m_1 and m_2-vectors, x and y are n and n_o-vectors respectively, and rank $(A_1) = m_1$ $(\leq n)$. The *process of elimination* for eliminating x from the given system is defined as:
1) Using the *general solution* to (A.2) for some fixed $p = \bar{p} \in \mathbb{R}^n$, solve for x:

$$x = A_1^\dagger (b_1 - A_2 y) + (\mathrm{I} - A_1^\dagger A_1)\bar{p} \, . \tag{A.4}$$

2) Substitute into (A.3) to obtain:

$$(A_4 - A_3 A_1^\dagger A_2) y = A_3 (\mathrm{I} - A_1^\dagger A_1)\bar{p} + (b_2 - A_3 A_1^\dagger b_1). \square \tag{A.5}$$

We note that when A_1 is square and nonsingular, $A_1^\dagger = A^{-1}$ and $\mathrm{I} - A_1^\dagger A_1 = 0$. Thus, both A_1 and A_1^{-1} are one-to-one linear mappings and $R(A_1) = \mathbb{R}^n$. Then, the above process is the well known variant of the Gauss–Jordan elimination referred to as "block elimination" or "block pivoting" (Dantzig [25]). Such elimination schemes have been used in conjunction with various decomposition or partitioning methods (e.g. [4, 26, 27, 28, 29, 30, 21]).

We let $Q \subset \mathrm{R}^{n+n_o}$ be the $(n + n_o - m_1)$-dimensional subspace parallel to (A.2) and let

$$\bar{Q} = \{(x, y) \in \mathbb{R}^{n+n_o} \mid x + A_1^\dagger A_2 y = 0\} \subseteq Q$$

$$Q_1 = \{(x, y) \in \mathbb{R}^{n+n_o} \mid x = 0\}$$

$$\bar{Q}^\perp = Q_2 = \{(x, y) \in \mathbb{R}^{n+n_o} \mid y = (A_1^\dagger A_2)' x\}$$

$$Q_3 = \{(x, y) \in \mathbb{R}^{n+n_o} \mid y = 0\}$$

where $\dim \bar{Q} = \dim Q_1 = n_o$ and $\dim Q_2 = \dim Q_3 = n$.

Lemma A.1: The process of elimination is equivalent to a nonorthogonal projection operation projecting the normals to the m_1 hyperplanes defined by (A.3) onto Q_1 *along* Q_2.

Proof: Consider the homogeneous system (A.2)–(A.3)

$$By = \begin{pmatrix} A_1 & A_2 \\ A_3 & A_4 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = 0 \tag{A.4}$$

Since rank $(A_1) = m$ $(\leq n_o)$, we can define the matrix operator

$$T = \begin{pmatrix} 0 & 0 \\ -(A_1^\dagger A_2)' & \mathrm{I} \end{pmatrix} \tag{A.5}$$

where $(A_1^\dagger A_2)'$ is an (n_o, n)-submatrix of T. We note that the range and the null space of T have the property that

$$R(T) = \left\{ (x, y) \in \mathrm{R}^{n+n_o} \;\middle|\; T\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} \right\} =$$

$$= \{(x, y) \in \mathrm{R}^{n+n_o} \mid x = 0\} \equiv Q_1 \tag{A.6.1}$$

$$N(T) = \left\{ (x, y) \in \mathrm{R}^{n+n_o} \;\middle|\; T\begin{pmatrix} x \\ y \end{pmatrix} = 0 \right\} =$$

$$= \{(x, y) \in \mathrm{R}^{n+n_o} \mid y = (A_1^\dagger A_2)'x\} \equiv Q_2. \tag{A.6.2}$$

Furthermore, $T^2 = T$ and therefore (*Thm.* C.12.9, pp. 256 in Zadeh – Desoer [25]), T projects on Q_1 along Q_2. Now apply the projection operation to the normals of the manifolds (A.2)–(A.3) to obtain:

$$TB' = \begin{pmatrix} 0 & 0 \\ 0 & (A_4 - A_3 A_1^\dagger A_2)' \end{pmatrix} \tag{A.7}$$

and compare the result to *definition A.2.* □

Similarly, $T^c = \mathrm{I} - T$ is the projection operator projecting any $g = (\bar{x}, \bar{y}) \in \mathbb{R}^{n+n_o}$ onto Q_2· along Q_1 since $Q_1 \oplus Q_2 = \mathbb{R}^{n+n_o}$ and for $u \in Q_1$, $v \in Q_2$, $T^c u = (\mathrm{I}-T)\, u = 0$; $T^c v = (\mathrm{I}-T)\, v = v$ and $(T^c)^2 = (\mathrm{I}-T)(\mathrm{I}-T) = \mathrm{I} - 2T + T^2 = T$. It is also interesting to note that T' projects any $g \in \mathbb{R}^{n+n_o}$ onto Q along Q_3 and $T^{c\,\prime} = T'^{c} = \mathrm{I} - T^c$ projects any g onto Q_3 along Q.

The orthogonal projection operator

$$P = \mathrm{I} - \begin{pmatrix} A_1' \\ A_2' \end{pmatrix} \left((A_1 \quad A_2) \begin{pmatrix} A_1' \\ A_2' \end{pmatrix} \right)^{-1} (A_1 \quad A_2) \tag{A.8}$$

projects any point $g \in \mathbb{R}^{n+n_o}$ onto Q and consequently $P^c = \mathrm{I} - P$ projects any g onto Q_1. The operators P and P^c are of particular interest in *GPM* (Rosen [3]). For the class of large scale problems studied in this paper, P may *not* be easily computed. However, the following two results establish means by which the projection of a point under P can be measured in terms of its projection under T.

As an illustration of these projection operations, consider the affine hyperplane: $x + 2y = 0$ where $(x, y) \in \mathbb{R}^2$, $m_1 = 1, n = n_o = 1$. Then, $A_1^\dagger = 1, A_1^\dagger A_2 = 2,$

$$T = \begin{pmatrix} 0 & 0 \\ -2 & 1 \end{pmatrix} \; ; \; T^c = \mathrm{I} - T = \begin{pmatrix} 1 & 0 \\ 2 & 2 \end{pmatrix}$$

$$P = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 2 \end{pmatrix} \left((1, 2) \begin{pmatrix} 1 \\ 2 \end{pmatrix} \right)^{-1} (1, 2) = (1/5) \begin{pmatrix} 4 & -2 \\ -2 & 1 \end{pmatrix}$$

$$P^c = (1/5) \begin{pmatrix} 1 & 2 \\ 2 & 4 \end{pmatrix} .$$

Then, the point $g = (2, 1)$ has projections: $Tg = (0, 3)$; $T^c g = (2, 4)$; $T'g = (-2, 1)$; $T^{c\prime} g = (0, 4)$; $Pg = 3/5(2, -1)$ and $P^c g = 1/5(4, 8)$ (fig. A.1).

Lemma A.2: Let rank $(A_1) = m_1$ and let P and T be matrix operators defined by (A.8) and (A.5) respectively. Then, for any point $g = (x, y) \in \mathbb{R}^{n+n_o}$, $PTg = Pg$.

Proof: First we note that the (m_1, m_1)-order inverse

$$\left((A_1 \;\; A_2) \begin{pmatrix} A_1' \\ A_2' \end{pmatrix} \right)^{-1} = (\mathrm{I} + K^{-1} A_2 A_2')^{-1} K^{-1} = M^{-1} K^{-1}$$

where $K^{-1} = (A_1 A_1')^{-1}$ exists since rank $(A_1) = m_1$. Furthermore, $P^2 = P$, $P' = P$ and $T' = T$ but $T' \neq T$. Since both P and T are projection operators it suffices to show $PT = P$. Consider

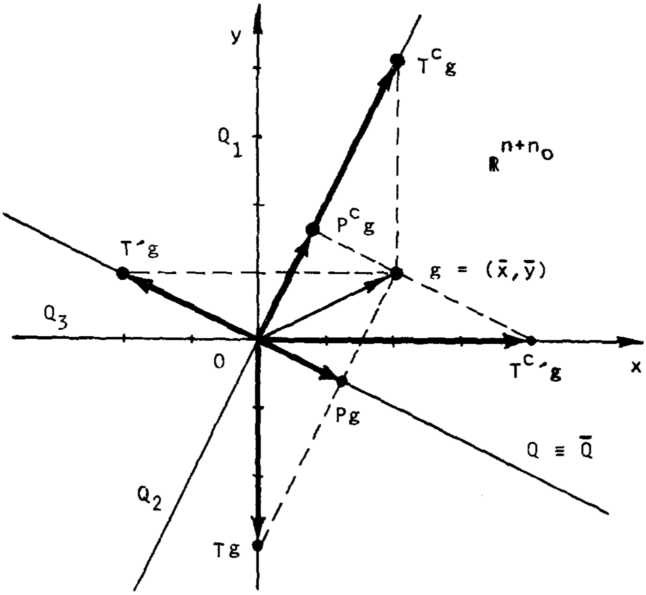

Fig. A.1. Projections under T, T^c, T', P and P^c.

$$P(T - \mathrm{I}) = \left(\begin{pmatrix} \mathrm{I} & 0 \\ 0 & \mathrm{I} \end{pmatrix} - \begin{pmatrix} A_1' \\ A_2' \end{pmatrix} M^{-1}K^{-1}(A_1 \quad A_2)\right)\begin{pmatrix} -\mathrm{I} & 0 \\ -A_2'K^{-1}A_1' & 0 \end{pmatrix}$$

$$= \begin{pmatrix} -\mathrm{I} & 0 \\ -A\,KA' & 0 \end{pmatrix} + \begin{pmatrix} A' \\ A' \end{pmatrix} M^{-1}K^{-1}(A_1 + A_2A_2'K^{-1}A_1 \quad 0) =$$

$$= \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}$$

Clearly, $S_{12} = 0$ and $S_{22} = 0$. Furthermore,

$$S_{11} = -\mathrm{I} + A_1'M^{-1}K^{-1}(\mathrm{I} + A_2A_2'K^{-1})A_1 = \mathrm{I} + A_1'M^{-1}K^{-1}M'A_1$$

since $M' = (\mathrm{I} + A_2A_2'K^{-1})$. Postmultiplying by $A_1'K^{-1}$ and premultiplying by $K^{-1}A$, we have: $K^{-1}A_1S_{11}A_1'K^{-1}A_1 = -K^{-1} + M^{-1}K^{-1}M'$. Postmultiplying this equation by M and using the symmetry of M and K gives: $MK^{-1}A_1S_{11}A_1'K^{-1}A_1 = -MK^{-1} + K^{-1}M' = 0$. By assumption,

M and K are nonsingular. Thus, $A_1 S_{11} A' = 0$, which in view of the linear independence of the columns of A_1' can only be true when $S_{11} = 0$. To show that $S_{21} = 0$, consider

$$S_{21} = -A_2'K^{-1}A_1' + A_2'M^{-1}K^{-1}(A_1 + A_2A_2'K^{-1}A_1) =$$

$$= -A_2'K^{-1}A_1' + A_2'M^{-1}K^{-1}M'A_1$$

and postmultiplying by $A_1'K^{-1}$

$$S_{21}A_1'K^{-1} = -A_2'K^{-1} + A_2'M^{-1}K^{-1}M' =$$

$$= A_2'(-K^{-1} + M^{-1}K^{-1}M') = A_2'M^{-1}(-MK^{-1}K^{-1}M') = 0$$

or $S_{21}A_1' = 0$ which requires that $S_{21} = 0$. □

Theorem A.3: Let $g \in \mathbb{R}^{n+n_o}$ be any given point. Then, $\|Pg\| \leq \|Tg\|$.

Proof: From *lemma A.2*, $\|Pg\| = \|PTg\| \leq \|P\|\|Tg\| \leq \|Tg\|$ since $\|P\| = \sup_{\|g\|=1} \|Pg\| = 1$. □

The importance of this theorem is obvious. For a given $\epsilon > 0$, if $\|Tg\| < \epsilon$, then $\|Pg\| < \epsilon$ is established without explicit knowledge of P.

Appendix 2:

A simple numerical example

We consider the problem P: min $\{f(x, y) \mid B'x + D'y \leq h\}$ where $x \in \mathbb{R}^2$, $y \in \mathbb{R}^1$ and $f(x, y) = cx + \frac{1}{2}xMx + xPy + dy + \frac{1}{2}yNy$. $c = (-1/4)(8, 7)$; $P' = (1, 1)$; $d = -7/4$; $N = 2$;

$$M = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}; \quad B' = \begin{pmatrix} 1 & 1 \\ -1 & 0 \\ 0 & -1 \\ 0 & 0 \end{pmatrix}; D' = \begin{pmatrix} 1 \\ 0 \\ 0 \\ -1 \end{pmatrix}; h = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

We define

$$f(x^t + v, y^t + w) = f(x^t, y^t) + (cv + \tfrac{1}{2} vMv + vPw + dw + \tfrac{1}{2} wNw) +$$

$$+ x^t(Mv + Pw) + y^t(Nw + P'v)$$

where v, w represent the change in x, y respectively.

We start with the "feasible" $y^o = 3/4$.

PI^o: $\min \left\{ f(x, y^o) \;\middle|\; \begin{array}{l} (x)_1 + (x_2) \leq 1 - y^o \\ (x)_1, (x)_2 \geq 0 \end{array} \right\}$

gives $\bar{x}^o = (1/4, 0); \underline{I}^o = \{1, 3\}$, and we compute $f(\bar{x}^o, y^o) = -1$

$$R^{o\,\prime} = \begin{pmatrix} 1 & 0 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

$$v = -\begin{pmatrix} 1 \\ 0 \end{pmatrix} w\,.$$

Substituting this into $f(\bar{x}^o + v, y^o + w)$ and into the constraints we obtain PII^o: $\min \{F(w) \mid -3/4 \leq w \leq 0\}$ with $F(w) = f(\bar{x}^o, y^o) + \frac{3}{4} w + (w)^2$ which has an interior minimum at $w^o = -3/8$. Thus,

$$f(x^1, y^1) = F(w^1) = -1 - 9/32 + 9/64 = -73/64 = -292/256$$

and since $v^1 = -(1, 0)(-3/8) = (3/8, 0)$, we obtain

$$(x^1, y^1) = (1/4, 0, 3/4) + (3/8, 0, -3/8) = (5/8, 0, 3/8).$$

PI^1: $\min \left\{ f(x, y^1) \;\middle|\; \begin{array}{l} (x)_1 + (x)_2 \leq 1 - y^1 = 5/8 \\ (x)_1, (x)_2 \geq 0 \end{array} \right\}$

gives $\bar{x}^1 = (7/16, 3/16) \neq x^1; \underline{I}^1 = \{1\}$, and we compute

$$R^{1\,\prime} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \left((1 \;\; 1) \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right)^{-1} \cdot 1 = (1/2) \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$v = (-1/2) \begin{pmatrix} 1 \\ 1 \end{pmatrix} w.$$

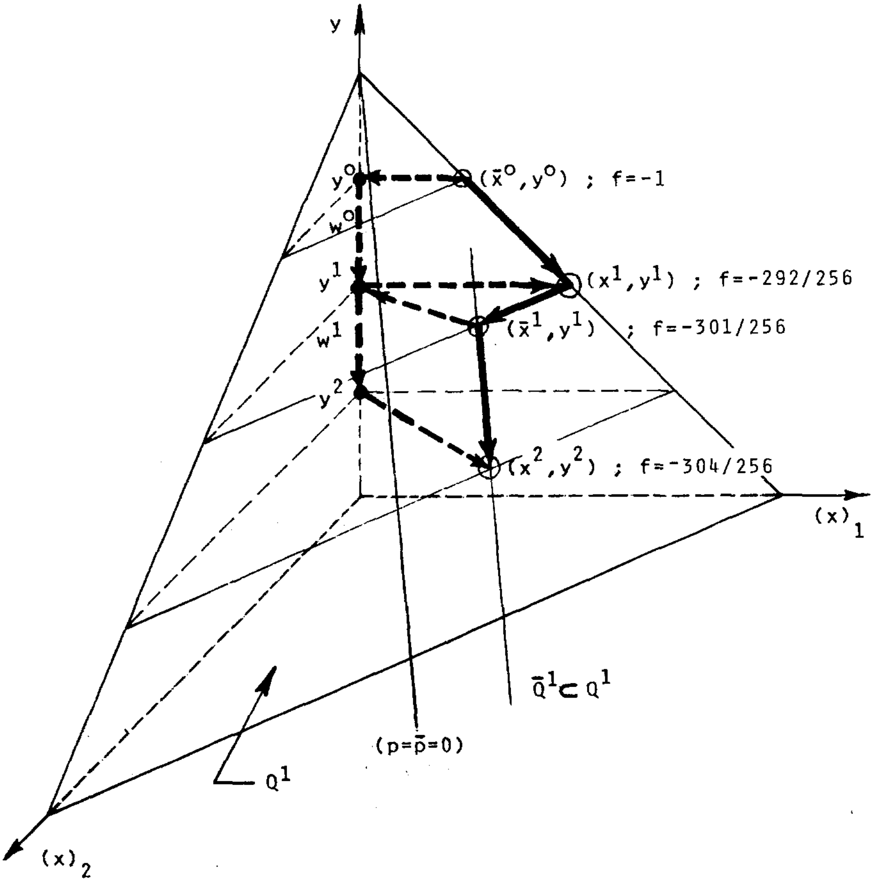
y
$(\bar{x}^o, y^o)$; f=-1
y^o
w^o
y^1
(x^1, y^1) ; f=-292/256
$(\bar{x}^1, y^1)$; f=-301/256
w^1
y^2
(x^2, y^2) ; f=-304/256
$(x)_1$
$\bar{Q}^1 \subset Q^1$
$(p=\bar{p}=0)$
Q^1
$(x)_2$

Fig. A.2. Solution path for numerical example.

The function value is $f(\bar{x}^1; y^1) = -301/256$. Substituting into f and constraints, we obtain

$PII^1 : \min\{F(w) \mid -3/8 \leq w \leq 3/8\}$

with

$$F(w) = f(\bar{x}^1, y^1) + (3/16)w + (3/4)(w)^2$$

which has an interior minimum $w^1 = -1/8$. Thus,

$$f(x^2, y^2) = F(w^1) = -301/256 + (3/16)(-1/8) +$$

$$+ (3/4)(1/64) = -304/256$$

and since $v^1 = -(1/2)(1,\ 1)(-1/8) = (1/16,\ 1/16)$, we obtain

$$(x^2, y^2) = (7/16,\ 3/16,\ 3/8) + (1/16,\ 1/16,\ -1/8) =$$

$$= (1/2,\ 1/4,\ 1/4)$$

$$PI^2\colon \min \left\{ f(x,y) \;\middle|\; \begin{array}{l} (x)_1 + (x)_2 \leq 1 - y^2 = 3/4 \\ (x)_1, (x)_2 \geq 0 \end{array} \right\}$$

gives $\bar{x}^2 = (1/2,\ 1/4) = x^2$; $\underline{I}^2 = \{1\}$; with

$$u^2 = (-1/2)(1,\ 1)\ \nabla_x f(x^2, y^2) = (-1/2)(1,\ 1) \begin{pmatrix} -1/2 \\ -1/2 \end{pmatrix} =$$

$$= 1/2 \geq 0\,.$$

Thus, we conclude that $(x^*, y^*) = (1/2,\ 1/4,\ 1/4)$ solves P. The solution path is shown in fig. A.2. Note that the direction of the step $((x^2, y^2) - (\bar{x}^1, y^1))$ is parallel to the line denoted by $p = \bar{p} = 0$.

References

[1] A.M. Geoffrion, "Elements of large scale mathematical programming", *Management Science* 16, No. 11 (1970) 652–675.

[2] G.B. Dantzig and P. Wolfe, "Decomposition principle for linear programs," *Operations Research* 8 (1960) 101–111; Also see
G.B. Dantzig and P. Wolfe, "The decomposition algorithm for linear programs," *Econometrica* 29 (1961) 767–778.

[3] J.B. Rosen, "Partition programming," *American Mathematical Society – Notices* 7, Abstract No. 572–22 (1960) 718–719.

[4] J.B. Rosen, "Convex partition programming," in: Eds. R.L. Graves and P. Wolfe, *Recent advances in mathematical programming,* McGraw-Hill, New York, 1963, pp. 159–176.

[5] O.L. Mangasarian, *Nonlinear programming* (McGraw-Hill, New York, 1969).

[6] M.D. Grigoriadis, "A projective method for a class of structured nonlinear programming problems," IBM New York Scientific Center Report No. 320–2992 (1970).

[7] H.W. Kuhn and A.W. Tucker, "Nonlinear programming," *Proceedings of the second Berkeley Symposium of mathematical statistics and probability* (University of California Press, Berkeley, California, 1951).

[8] J.B. Rosen, "The gradient projection method for nonlinear programming, Parts I and II" *SIAM Journal on Applied Mathematics,* 8, No. 1 (1960) pp. 181–217; and 9, No. 4 (1961) 514–532.

[9] D. Goldfarb, "Extension of Davidon's variable metric method to maximization under linear inequality and equality constraints," *SIAM Journal on Applied Mathematics,* 17, No. 4 (1969) 739–764.

[10] G. Zoutendijk, *Methods of feasible directions* (Elsevier Publishing Company, Amsterdam, 1960).

[11] W.I. Zangwill, *Nonlinear programming – a unified approach* (Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1969).

[12] O.L. Mangasarian, "Equivalence in nonlinear programming," *Navals Research Logistics Quarterly* 10, No. 4 (1963) 299–306.

[13] G.B. Dantzig, *Linear programming and extensions* (Princeton University Press, Princeton, N.J., 1963).

[14] D.A. D'Esopo, "A convex programming procedure", *Naval Logistics Research Quarterly* 6, No. 1 (1959) 33–42.

[15] D.J. Wilde, *Optimum seeking methods* (Prentice-Hall, Englewood Cliffs, New Jersey, 1964).

[16] J. Warga, "Minimizing certain convex functions," *SIAM Journal on Applied Mathematics,* 16, No. 1 (1960) pp. 588–593.

[17] R. Hooke and T.A. Jeeves, "Direct search' solution of numerical and statistical problems," *Journal of the Association for Computing Machinery* 8, No. 2 (1961) 212–229.

[18] P. Fauré and P. Huard, "Résolution des programmes mathématiques a fonction nonlinéaire par la méthode du gradient réduit", *Revue Française de Recherche Opérationnelle* 9 (1965).

[19] J.B. Rosen, Private communication (January 1970).

[20] J.B. Rosen and S. Suzuki, "Construction of nonlinear programming test problems," *Communications of the Association for Computing Machinery* 8, No. 2, (1965) pp. 113.

[21] M.D. Grigoriadis, "A dual generalized upper-bounding technique", *Management Science,* 17, No. 5 (1970) 269–284.

[22] E.H. Moore, "General analysis (part 1)," *Memoirs of the American Philosophical Society,* Vol. 1 (1935).

[23] R. Penrose, "A generalized inverse for matrices," *Proceedings of the Cambridge Philosophical Society,* 51 (1955) 406–413.

[24] T.N.E. Greville, "The pseudoinverse of a rectangular or singular matrix and its application to the solution of systems of linear equations," *SIAM Review* 1, No. 1 (1959) 38–43.

[25] G.B. Dantzig, "Upper bounds, secondary constraints and block triangularity in linear programming," *Econometrica* 23, No. 2 (1955) 174–183.

[26] J.B. Rosen, "Primal partition programming for block diagonal matrices" *Numerische Mathematik* 6 (1964) 250–260.

[27] K. Ritter, "Structured quadratic programming problems," Mathematics Research Center Report No. 718, The University of Wisconsin, Madison, Wisconsin (January 1967)..

[28] G.B. Dantzig and R.M. Van Slyke, "Generalized upper bounding techniques", *Journal of Computer and System Sciences,* 1 (1967) 213–226.

[29] R.N. Kaul, "An extension of generalized upper-bounded techniques for linear programming," Operations Research Center Report No. 1965–27, University of California, Berkeley, California (1965).

[30] M.D. Grigoriadis and K. Ritter, "A decomposition method for structured linear and nonlinear programs", *Journal of Computer and System Sciences* 3, No. 4 (1969) 335–360.