Large-Scale Decomposition for Successive Quadratic Programming

by

S. Vasantharajan and L. Biegler

EDRC-06-32-87

LARGE-SCALE DECOMPOSITION FOR SUCCESSIVE QUADRATIC PROGRAMMING

by

S. Vasantharajan and L. T. Biegler

Department of Chemical Engineering
Carnegie-Mellon University
Pittsburgh, Pa 15213

Table of Contents

List of Figures

List of Tables

Successive Quadratic Programming (SQP) has emerged as the algorithm of choice for solving moderately-sized nonlinear process optimization problems. However, as the nonlinear programming problem becomes large (over 100 variables, say) storage requirements for the Hessian matrix and the computational expense of solving large quadratic programs can become prohibitive. To overcome this problem, Westerberg and coworkers proposed two SQP decomposition strategies in 1980 and 1983. The first strategy overcomes this problem but is difficult to implement, while the second has been observed to give inconsistent results. The strategy in this paper uses range and null space projections to develop a decomposition algorithm that is both easy to implement and performs as well as the full SQP algorithm on small problems. This range and null space decomposition (RND) allows for sparse implementations and thus solves large problems easily and reliably. Theoretical development of the RND method is presented as well as a geometric interpretation of this approach compared to others. Finally, a thorough numerical comparison of SQP strategies is presented on a battery of nonlinear programming and process optimization test problems.

1. Introduction

Successive Quadratic Programming (SQP) (Han (1977), Powell (1977)) has been used in numerous applications for chemical engineering optimization problems. For optimization of complex, computationally intensive models such as process flowsheets (Lang and Biegler (1987), Chen and Stadtherr (1985), Kisala (1985), Evans et al. (1985)), with a small to moderate number of variables, SQP consistently requires fewer function evaluations than other optimization algorithms. SQP can be derived as a Newton method for solving the optimally conditions. The basic step in the SQP algorithm is the formulation and solution of the Quadratic Programming Problem (QPP). Aside from the effort required for function and gradient evaluations, this is the most time consuming step for the algorithm. This operation requires the storage and update of a Hessian matrix at each iteration, which is of the size of the number of process variables at each iteration. Hence, as the optimization problem becomes large (over 100 variables, say), storage requirements for the Hessian matrix as well as the computational expense involved in solving quadratic programs can become prohibitive.

In most optimization problems only the Hessian matrix projected into the null space at the solution is positive-definite. The entire Hessian matrix, on the other hand, is generally not positive-definite, especially if the optimization problem is large and sparse. Consequently,

approximating the Hessian matrix by a dense, positive-definite updating formula may not lead to a fast rate of convergence. Boggs et al. (1982) proved that SQP converges at a superlinear rate if both the Hessian matrix and its quasi-Newton approximation are positive-definite. On the other hand, Powell (1978a) showed that when the Hessian is not positive-definite the rate of convergence deteriorates to one termed "two-step superlinear". This deterioration adds further incentive to developing a strategy that updates the *projected* Hessian matrix instead of the full matrix.

Over the past few years, several decomposition algorithms have been proposed that reduce the size of the QPP by eliminating dependent variables and equality constraints. For process optimization problems, Westerberg and co-workers proposed decomposition strategies in 1980 and 1983. The first, by Berna, Locke and Westerberg (1980) (BLW), is an extension of Poweirs SQP algorithm (1977), and can be be derived through block Gaussian elimination applied to the optimality conditions of the QPP at each step. The resulting reduced QPP retains the symmetric structure of the original QPP and leads to the solution of the same problem but with a considerable reduction in core storage of Hessian matrix. Here, if the Jacobian matrix for the dependent variables is non-singular, and if there is enough storage for the Hessian update vectors, this approach can be shown to be equivalent to the original, undecomposed (or full) SQP algorithm. However, this approach is difficult to implement and because a Hessian matrix is constructed in the space of all the process variables and then projected into the subspace of the decisions, storage requirements of the rank 2 updates may still be prohibitive with a large number of SQP iterations. This approach was implemented by Chen and Stadtherr (1985) and by Jackson, Hutchison and Morton (1986), who noted the above-mentioned points. Also since the full Hessian is updated, in principle, a dense, positive-definite approximation is made to a matrix that generally has neither of these properties.

The difficulties mentioned above prompted Locke, Edahl and Westerberg (1983) to propose a new decomposition strategy (LEW), which computes and stores a reduced Hessian directly. Derived through sensitivity analysis, variables are partitioned into two sets, the decisions and the dependents, and a QPP is constructed only in the space of the decision variables. The reduced Hessian is directly estimated in the space of these variables and the reduced QP yields a search direction for the decision variables. The search direction for the dependent variables is then found that satisfies the set of linearized equality constraints. Chan and Prince (1986)

reported excellent results with the LEW decomposition on a series of Jiomogeneous flowsheets with mostly linear mass balance models. Trevino-Lozano (1985) and Kisala (1985), on the other hand, mentioned that the LEW decomposition usually requires more iterations than the full SQP method and in some instances the LEW decomposition leads to inconsistent QP's at infeasible starting points. To remedy this situation, they suggest converging. the equality constraints first with Newton's method.

Thus for large optimization problems solved by SQP, the BLW strategy is difficult to implement and may require a large amount of storage, while the LEW approach gives inconsistent results. In this paper, we present a strategy which overcomes both of these problems. It is based on range and null space methods, recently developed by Nocedal and Overton (1985) for reduced SQP, and uses orthogonal (but not orthonormal) bases for decomposition. This method can be shown to have the same convergence rate as the full (undecomposed) SQP method, but requires only the calculation and storage of the null-space projection (2-sided projection) of the Hessian matrix. Using this projected update was first suggested by Murray and Wright (1978), but most of these algorithms developed in the optimization literature use orthonormal null space factorizations based on *QR* decomposition. The algorithm presented in this paper extends the Nocedal and Overton approach to systems with inequality and equality constraints and allows the solution of large, sparse optimization problems by using sparse, nonorthonormal factorizations.

In the subsequent sections, we develop this strategy, first for an orthonormal projection, and then for nonorthonormal bases. Detailed derivation and outline of the above algorithm as implemented are also given. In addition, a theoretical justification of the effectiveness of this nonorthonormal strategy is presented and a geometric interpretation is given to compare the range and null space strategy to previous strategies, including the original, undecomposed SQP algorithm. Finally, a battery of nonlinear programming and process optimization test problems is solved and a numerical comparison is presented to demonstrate the effectiveness of this approach.

2. Range And Null Space Decomposition [RND]

The general process optimization problem can be stated as follows:

$$\text{(NLP)} \qquad \begin{aligned} Min \quad & \varphi(z) \\ \text{s.t.} \quad & g(z) \le 0 \\ & h(z) = 0 \end{aligned}$$

where

φ: $\mathbb{R}^n \to \mathbb{R}$ - objective function

g: $\mathbb{R}^n \to \mathbb{R}^r$ - inequality constraints

h: $\mathbb{R}^n \to \mathbb{R}^m$ - equality constraints

$z \in \mathbb{R}^n$ - set of variables

SQP employs a modified quasi-Newton method to converge to the optimality conditions of the NLP by solving the following quadratic program (QP):

$$\text{(QP1)} \qquad \begin{aligned} \underset{\overline{p}}{Min} \quad & \nabla\varphi(z_k)^T \overline{p} + \tfrac{1}{2} \overline{p}^T B \overline{p} \\ \text{s.t.} \quad & h(z_k) + \nabla h(z_k)^T \overline{p} = 0 \\ & g(z_k) + \nabla g(z_k)^T \overline{p} \le 0 \end{aligned}$$

where

$\overline{p}$ - search direction

B - BFGS update to the Hessian of the Lagrangian

The first order necessary conditions for QP1 can be represented in matrix form as follows:

$$\begin{bmatrix} B & \nabla g_A & \nabla h \\ \nabla g_A^T & 0 & 0 \\ \nabla h^T & 0 & 0 \end{bmatrix} \begin{bmatrix} \overline{p} \\ u \\ \overline{v} \end{bmatrix} = - \begin{bmatrix} \nabla\varphi \\ g_A \\ h \end{bmatrix}$$

$$z_{k+1} = z_k + \overline{p}$$

where,

$$g_A(z_k) + \nabla g_A^T \overline{p} = 0, \qquad g_j(z_k) + \nabla g_j^T \overline{p} < 0, \quad \forall\, j \notin A \tag{1}$$

Let z be a given point with V/7, the (n x m) matrix of equality constraint gradients (inequality constraints known to be active can be included in *h)* having full column rank, i.e., rank m. Let $\bar{Z}(z)$ be an [n x (n-m)] matrix with orthonormal columns spanning the null space of $V/?^T$ and let $\bar{Y}(z)$ be an (n x m) matrix with columns forming an orthonormal basis for the range space of V/7. One very useful way of obtaining $\bar{Z}$ is by forming the *OR* factorization:

$$V/7 = \begin{bmatrix} \bar{Y}(z) & \bar{Z}(z) \end{bmatrix} \begin{bmatrix} \bar{R}(z) \\ 0 \end{bmatrix} \tag{2}$$

where $\bar{/?}$ is a (m x m) upper triangular matrix and Q is given by:

$$Q = \begin{bmatrix} \begin{bmatrix} \bar{Y} & \bar{Z} \end{bmatrix} & 0 \\ 0 & I \end{bmatrix} \tag{3}$$

an orthogonal matrix of order (n+m). Of course,

$$\begin{aligned} \bar{Z}^T \bar{Z} &= I & \bar{Y}^T \bar{Y} &= I \\ \bar{/}^T \bar{Z} &= 0 & V/7^T \bar{Z} &= 0 \end{aligned} \tag{4}$$

Then eqn. 1 can be written as

$$Q^T \begin{bmatrix} B & Vg_A & V/7 \\ ** & 0 & 0 \\ V/7^T & 0 & 0 \end{bmatrix} QQ^T \begin{bmatrix} \bar{p} \\ u \\ V \end{bmatrix} = -Q^T \begin{bmatrix} \nabla\varphi \\ g_A \\ h \end{bmatrix}$$

$$z_{k+1} = z_k + \bar{p} \tag{5}$$

i.e.,

$$\begin{bmatrix} \bar{Y}^T B \bar{Y} & \bar{Y}^T B \bar{Z} & \bar{Y}^T Vg_A & \bar{R} \\ \bar{Z}^T B \bar{Y} & \bar{Z}^T B \bar{Z} & \bar{Z}^T g_A & 0 \\ Vg_A^T \bar{Y} & Vg_A^T \bar{Z} & 0 & 0 \\ \bar{R}^T & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \bar{Y}^T \bar{p} \\ \bar{Z}^T \bar{p} \\ u \\ V \end{bmatrix} = - \begin{bmatrix} \bar{Y}^T \nabla\varphi \\ \bar{Z}^T \nabla\varphi \\ 9 \\ h \end{bmatrix} \tag{6}$$

Renaming $\bar{Y}^T \bar{p}$ as $\bar{p}_y$ and $\bar{Z}^T \bar{p}$ as $\bar{p}_z$, we know from orthonormal properties that

$$\bar{p} = \bar{V}\bar{p}_y + \bar{Z}\bar{p}_z ,$$

$$z_{k+1} = z_k + \bar{p} \qquad (7)$$

Thus, the system to be solved can be represented as follows:

a. $R\,\bar{p}_y = -/7,$

b. $(Z\,B\,Z)\,\bar{p} = -\,Z\,V<p - Z\,Vg\;\; u - Z\,B\,Y\,\bar{p}\;\;,$

c. $\bar{p} = \bar{Y}\,\bar{p}_y + \bar{Z}\,\bar{p}_z ,$

d. $z_{k+1} = z_k + J\tilde{f} ,$

e. $R\,7_{k+j} = -\,7\,V\,\wedge - /\,V\,f\,f\;\;(/ - K \ll p$ (8)

As the algorithm converges, $p_z \to 0$ and eqn. e can be simplified to:

$$R(z_{k+1})\,\bar{V}_{k+1} = -\,Y(z_{k+1})\left[\nabla\phi(z_{k+1}) + \nabla g_A\, u\right] \qquad (9)$$

which is a "least squares" multiplier estimate, and is based on the rationale that a "first-order" estimate given above, obtained from the new information at the current point, should be preferable to using the "second-order" estimate eqn. e which is just the Newton prediction of values at the current point, based on the information at the previous iteration. Wright (1976) shows that using eqn. 9 instead of eqn. e also yields quadratic convergence of the method. Furthermore, Tapia (1977) shows that use of eqn. 9 gives a method which is Q-quadratically convergent in z, in the sense of Ortega and Rheinboldt (1970).

The second reduction is done by dropping the $(Z\,B\,Y\,p_y)$ part from eqn. b. If the constraints are linear, the range space of $V/7^T$ is a constant, and the term $(Z\,B\,Y)$ is irrelevant, since given an initial feasible point, all subsequent iterates can be chosen to be feasible, and hence there is no range space movement Under these circumstances, updating an approximation to $(Z\,B\,Z)$ is preferable to updating the full Hessian. The extension of this concept to the nonlinear case, suggested by Murray and Wright (1978) is motivated primarily for the purpose of reducing the dimension of the approximating matrix, which can be updated by DFP or BFGS methods. Note that the matrix $(Z\,B\,Z)$ is generally dense and positive-definite and thus is suited for these updating formulae.

The convergence analysis of such an algorithm, which is obtained by ignoring the projection of the Hessian in the range space, can be intuitively explained by drawing a parallel with the method proposed by Powell (1978a) in modifying Han's algorithm. The only difference between these two approaches is that Powell maintains an approximation to the true Hessian. However only the *(Z B Z)* portion of *B* should be expected to have any relation to the true Hessian matrix, since the former is positive-definite and the true Hessian, in general is not Thus while the range space projection term is discarded in this algorithm, it is approximated by an inaccurate term in Powell's algorithm and hence a similar convergence property, viz. a two-step Q-superlinear convergence rate, can be expected, *i.e.*,

$$\|z_{k+1} - z^*\| \le a_k \|z_k - z^*\|, \quad \text{with } a_k \to 0 \text{ as } k \to \infty \tag{10}$$

A rigorous proof of the convergence rate is given by Nocedal and Overton although an intuitive explanation for the same can be given as follows: Let z be the current iterate. Solving the system of equations 8, in the absence of the *(Z B Y)* term or with any inaccurate approximation cannot be expected to give a point z_k, for which the error $\|z_k - z^*\|$ is smaller than $\|z_{k-1} - z^*\|$. However, the uncorrupted movement $\bar{p}_y$ in the range space would give an accurate step to the constraints, giving a value $\|\bar{p}_k\|$ which is relatively small compared to $\|z_{k-1} - z^*\|$. On the next iteration, the error incurred by dropping the second order information in $\bar{Y}$ is proportional to $\|\bar{p}_k\|$. so that the next iteration point generated, z_{k+1}, will have an error $\|z_{k+1} - z^*\|$ which is small compared to $\|z_{k-1} - z^*\|$ and hence a two-step Q-superlinear convergence property.

The final QP and steps involved in this algorithm employing orthonormal bases can be represented as follows:

$$\text{(QP2)} \qquad \begin{aligned} \textit{Min} \quad & \nabla f^T \bar{Z} p_z + \tfrac{1}{2} p_z^T \bar{Z}^T B \bar{Z} p_z \\ \text{S.t.} \quad & \nabla c_j^T \bar{Z} p_z \le -g + \nabla g^T \bar{Y} (\bar{R}^T)^{-1} h \end{aligned}$$

Solution of QP2 gives $\bar{p}_z$ and *u*. The other parameters viz. the range space movement and the multipliers for the equalities, *V* are computed as follows:

$$\bar{p}_y = -(\bar{R}^T)^{-1} h$$

$$V = -(\bar{R}^{-1})^T \bar{Y}^T (\nabla f + B p_z)$$

$$\bar{p} = \bar{Z}\,\bar{p}_z + \bar{/}\,\bar{p_y} \tag{11}$$

3. Extension To Nonorthonormal Bases

The focal point in the implementation of the Range and Null space method with orthonormal bases, is that in addition to the effort in function and gradient evaluation at each iteration, a *OR* factorization of the gradient matrix has to be performed to compute the orthonormal bases $\bar{Z}$ and $\bar{V}$. Although there are robust packages to do so, and it may not always be computationally prohibitive, a logical step in the improvement of this algorithm would be the use of alternate bases for range and null spaces, which would be easier to estimate and at the same time, retain all the desirable features of this algorithm.

In this section, we propose and detail a choice of orthogonal bases for the constraint matrix, and show that the derivation and conclusions of Nocedal and Overton are still valid. In the subsequent sections, we discuss a particular choice for the bases, which possesses the desirable orthogonal properties and which can be conveniently represented in terms of the gradients and not require much additional effort to compute. Further, we show that with such a choice the convergence property of the algorithm is not altered and 2-step Q-superlinear convergence is retained.

Let the new choice of bases for V/7, the constraint gradient matrix be expressed as linear combination of the orthonormal bases $\bar{Z}(z)$ and $\bar{Y}(z)$ as follows:

$$Y = \bar{Y}\,N \qquad Z = \bar{Z}\,M \tag{12}$$

where N G IR^{mxm} and M G JR^^^^^ and we still have the desirable property that:

$$Y^T Z = 0 \tag{13}$$

Define

$$V/?^T Y = V/?^T \bar{Y}N = R^1 = \bar{R}^T N \tag{14}$$

and set up (QP3) with the orthogonal bases and compare with the solution from the formulation with orthonormal bases viz. QP2.

(QP3) $\quad Min \quad V<f^T Z p_z + \backslash p\backslash (Z^T B Z) p_z$

$$\text{s.t} \quad Vg^1 \; Z \; p_z \leq -g + Vg^1 \; Y \; (R^1)^{-1} \; h$$

with

$$p_y = - (R^T Y^I \; h$$

$$v = - (R)^{-1} \; Y^T \; (\nabla \varphi + \nabla g \; u)$$

$$p = Z \; p_z + Y \; p_y \qquad (15)$$

Now let us compare the results from the steps in the Null and Range spaces and the overall estimates for the search direction and multipliers by substituting for Z and Y in terms of $\bar{Z}$ and $\bar{Y}$.

Null space: The result of QP3 gives the following:

$$(Z^T B \; Z) \; p_z =. - Z^T \; (V<p + V \wedge_A u)_9 \quad i.e.$$

$$M^1 \; (\bar{Z}^T B \tilde{Z}) \; M \; p_z = - M^7 \; Z^{-T} \; C7<p * Vg_A \; u), \quad i.e.$$

$$p_z = - (M)^{-1} \; (\bar{Z} \; B \; ZY^X \; Z^T \; C7<p + Vg_A \; u| \quad \text{or}$$

$$p_z = (MY' \; \bar{p}_z \qquad (16)$$

Making similar substitutions as done above for the null space in the expression for p_y and v_t it can be shown that for

Range space:

$$P_y = (V)^{-1} \; \bar{p}_y \qquad (17)$$

Multiplier:

$$v = V \qquad (18)$$

Search direction.

$$P = \bar{p} \qquad (19)$$

Here, although the range and null space directions are different, the reconstructed search direction is the *same* for either choice of bases. The same is true for the multiplier estimate.

4. ChQice Of Orthogonal Bases

In this section, having proved the validity of other orthogonal bases for the range and null space decomposition, we present specific choices for both, which are based on gradient information that is already calculated and are in addition sparse formulations, which greatly facilitate their calculation. Next we will compare this formulation with that of Locke, Edahl and Westerberg (LEW) and show that the latter is not a reduced orthogonal SQP formulation. A geometric representation of all the formulations will also be presented here.

Partition the set of variables z into dependents or pivoted $y \in JR^m$ and independent x G $1R^{1TM}$. The orthogonal bases can now be represented as follows:

$$Z = \begin{bmatrix} I \\ -(v_{y} Vr^{1} \; v/7^{T}_{A.} \end{bmatrix}$$

$$Y = \begin{bmatrix} \text{X,} & y \\ & I \end{bmatrix} \qquad (20)$$

whereupon the fact that Z is orthogonal to *Y* and to the gradient matrix $V/?^T$ can be verified by direct calculation. With this nonorthonormal choice for *Y,* it can be seen that the *R* matrix can be conveniently represented by $C7h^T\ Y)$. Thus, the null space of the constraint matrix V/? is defined in terms of a *partition of the variables,* rather than a matrix Z with orthonormal columns. The motivation for doing so is that an orthonormal Z or *Y* is not practical in terms of storage manipulation and computation for most large-scale problems. In addition, numerical difficulties are likely in the computation of the search direction *p* and the projected Hessian approximation (Gill et al. (1981)).

By substituting these values in QP3, the new system of equations can be written as

$$\text{(RND)} \qquad Mln \; [\, V \wedge^T - V_y{<}p^T \; C7_y h^T)^{'1} \; \nabla_x h^T \,] \; p_z + \tfrac{1}{2} p_z^T \left[Z^T B \; Z \right] p_z$$

$$\text{s.t} \quad \left[\nabla_x g^T - \nabla_y g^T \; (\nabla_y h^T)^{-1} \; \nabla_x h^T \right] p_z \leq - g + \left[V \quad V \; (\nabla_y h)^{-1} + V \right] p_y$$

with

$$p_y = - (V/?^T V)" h$$
$$v = - (V^T V/?)"' V^T (V<P * Vg u)$$
$$P = Z p_z + Y p_y \quad (21)$$

When V/7 has full column rank, the vector p_y is *unique,* since (V/?^{T} V) is a non-singular matrix. Defining a new matrix *a* as $[(\nabla_y Z?)^{-1} V_x/?^T "I$, it can be easily shown that the terms above can be conveniently represented as follows:

$$p_y = - (I + aaY (V \wedge V h.$$
$$V = - (V./?)" (I + aa^T)^{-1} Y^7 (\nabla\varphi + \nabla g u) \quad (22)$$

In order to show that the Locke, Edahl and Westerberg (LEW) decomposition is not a reduced orthogonal SQP method, it is essential to examine the differences between this formulation and the Range and Null space method described above. Designating the movement in the independent x variables as d_x and in the dependent *y* variables as d_y and using the definition of the basis Z as given before, the LEW$_1$ formulation can be expressed as follows:

(LEW) $\quad Min \quad V\wedge>^T Z d_{\%} + \_j d\wedge (Z^T B Z) d_{\%} [\nabla_y g^{d\%} (\nabla_y h^T)^{-1} h]$

s.L $\quad C7g^T Z) d_x <> -g \bullet$

with $\quad d_y = - (\nabla_y h^T)^{-1} (\nabla_x h^T d_x + h)$

$$v = - (\nabla_y h)^{-1} (\nabla_y \varphi + \nabla_y g u)$$

$$d = \begin{bmatrix} d_x \\ d_y \end{bmatrix} \quad (23)$$

Using the definition of the orthogonal basis Z it can be seen - that the reconstructed search direction *d* from LEW can be expressed as follows:

$$d = \begin{bmatrix} d_x \\ -(V/)"' [\nabla_x h^T d_X + *] . \end{bmatrix}$$

i.e. $\quad ^d = Z d_x + \begin{bmatrix} 0 \\ -(\nabla_y h^T)^{-1} h \end{bmatrix} \quad (24)$

Thus, net contribution from the null space for RND which is Zp_z is identical to that given by LEW which is represented as Zd_x. The essential difference is in the movement in the Range space, where, RND takes an orthogonal step to the tangent subspace of the active constraints, which is geometrically the shortest distance, unlike LEW. Now the range space movement in RND is given by Yp_y i.e. by -V [(I+saVCV $_y$7^1)"^{1} h 1] This cannot be represented by - [0 07'J?)'^{1} h\ toe range space contribution from LEW, if the orthogonal relation Z^T $Y(I+aa^T)\sim^l(V\ h_y^T)'^1$ h = 0 is expected to hold. If we were to substitute:

$$Y(I+aaY\ C7_u h^T Y^l = \begin{bmatrix} 0 \\ (\nabla_y h^T)^{-1} \end{bmatrix}$$

and premultiply both sides by Z^T, we have

$$Z^T\ Y(I+aa^T)^{-1}\ (V\ /\!>V = 0 \quad \text{and,}$$

$$Z^T \begin{bmatrix} 0 \\ <V^T)'' \end{bmatrix} = \begin{bmatrix} I & -\nabla_x h(\nabla_y h)^{-1} \end{bmatrix} \begin{bmatrix} 0 \\ (\nabla_y h^T)^{-1} \end{bmatrix}$$

$$= \begin{bmatrix} -\nabla_x h\ (\nabla_y h)^{-1}\ (\nabla_y h^T)^{-1} \end{bmatrix} \neq 0 \qquad (25)$$

in general. So the LEW decomposition is not a reduced orthogonal SQP method. Also, it can be seen that the two formulations are identical for a choice of [To I"] for the basis Y, which obviously does not satisfy the orthogonal requirement that $Y^TZ = 0$, and hence the LEW method is not a Range and Null space method.

The differences cited above, can be explained more lucidly by considering a geometrical representation of the solution paths of the two algorithms compared to that of the undecomposed SQP method. On a simple problem, which will be referred to as the *Example* problem, we see the following behavior:

Problem statement:

$$Min \quad [1 \quad 2] \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \tfrac{1}{2}\left[z_1^2 + z_2^2\right]$$

$$\text{S.L} \quad h: \ z_x + z_2 - 1 = 0$$

Parameters:

$$Vh = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad Y = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad \bar{Y} = (2)^{1/2} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$B = I \qquad Z = \begin{bmatrix} 1 \\ -1 \end{bmatrix} \qquad \bar{Z} = (2)^{-1/2} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$z_0 = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \qquad V(z) = \begin{bmatrix} 1 + z_1 \\ 2 + z_2 \end{bmatrix} \qquad h_0 = 1$$

$$\bar{R} = (2)^{1/2} \qquad R = 1$$

a) *Undecomposed QP solution:*

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ v \end{bmatrix} = - \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix}$$

i.e. $p_1 + v = -2$

$p_2 + v = -3$

$p_1 + p_2 = -1$

i.e. $p_1 = 0 \qquad p_2 = -1 \qquad v = -2$

i.e. $\bar{p} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$

b) *Orthonormal solution:*

$$[\bar{Z}' B Z]^{-1} \bar{p} = -\bar{Z}^T v^{\wedge} = (2)^{1/2}$$

i.e. $\bar{p}_z = (2)^{1/2}$ and $p_z = \begin{bmatrix} 1/2 \\ -1/2 \end{bmatrix}$

$\bar{p}_y = -(R)^{-T} \cdot 1, = -(2)^{-1/2}$ *i.e.,*

$$\bar{Y}\bar{p}_z = -\begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}$$

i.e. $$\bar{P} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

c) *Orthogonal solution:*

Let z_2 be the pivoted variable and z_1 the decision.

$$[Z^1BZ]p^{\wedge} = -Z^T \nabla\varphi = 1$$

i.e. $$p_z = 1/2 \quad \text{and} \quad Z p_z = \begin{bmatrix} 1/2 \\ -1/2 \end{bmatrix}$$

$$p_y = -(I + a\tilde{a}\dot{V} C7_y \bar{h}T \dot{h} = -1/2 \quad i.e.,$$

$$Y p_z = -\begin{bmatrix} 1/2 \\ 1/2 \end{bmatrix}$$

i.e. $$P = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

d) *Orthogonal solution:*

Let z_1 be the pivoted variable and z_2 the decision. This is equivalent to solving the same QP as done before with the new null space basis Z set to the negative of the old one.

$$[Z^T B Z_{\rfloor} I\, p_z = -Z^T \nabla\varphi = -1$$

I..* $$p_z = -1/2 \quad \text{and} \quad Z p_z = \begin{bmatrix} 1/2 \\ -1/2 \end{bmatrix}$$

i.e. the reconstructed movement in the null space is unchanged. Of course, the range space movement is unchanged as the basis *Y* is still the same and so the effective movement is still as the same as before.

e) *LEW solution:*

Let z_2 be the pivoted variable and z_1 the decision. As shown in eqn. 24, the null space movement Zd_x is the same as null space movement of die RND method, viz. Zp^ and the overall step d can be calculated as follows:

$$d = Zrf_x + \begin{bmatrix} 0 \\ -(V_y/?V\ h \end{bmatrix}$$

$$i.e. \quad d = Z\,p_z + \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

$$i.e. \quad d = \begin{bmatrix} i/2 \\ -3/2 \end{bmatrix}$$

f) *LEW solution:*

Let z_1 be the pivoted variable and z_2 the decision. Again from eqn. 24, the null space movement Zd_x is the same as null space movement of the RND method, viz. Zp_z and the overall step d can be calculated as follows:

$$d = Z\,d_x + \begin{bmatrix} -(\nabla_y h^T)^{-1}\ h \\ 0 \end{bmatrix}$$

$$i.e. \quad d = Z\,p_z + \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$$I.e. \quad tf = \begin{bmatrix} -1/2 \\ -1/2 \end{bmatrix}$$

Thus, not only is the step computed by the RND method independent of the manner of variable partitioning, but it also duplicates (in this case) the step computed by solving the undecomposed QP. The LEW, on the other hand, is sensitive to the choice of variable sets as pivoted and decisions, and because of a nonorthogonal movement in the range space produces different search directions. This nonoptimal movement to the tangent subspace of the active constraints can be explained as the source of inconsistent results which were observed by Trevino-Lozano (1985) and Kisala (1985) during the implementation of the LEW method.

Their remedy was to stay "close" to the constraint surface and thus keep the "range space" directions small. This is evident from graphical representation in fig 4-1 of the results computed in the *Example* problem.

5. Algorithmic Representation Of RND

In this section we define the algorithm, which is based on the Range and Null space method for the orthogonal choices of bases Z and *Y* as defined above by the system of equations (RND). We note that *a* is defined as $[(V_y h^T Y^1 V_x h^7 "]$, and *R* is defined as $[(I + a a^1) V_y hi.]$ Define a new vector *J3* as $[(V_y h^7)"* hi.]$ The matrix (I + *aa*\ can be seen from the definition of *a* to be of the same order of magnitude as the number of equality constraints eliminated, which is often a very significant number. This makes the L/U decomposition task, when inverting (implicitly) this matrix to compute $(R^7 Y\backslash$ computationally expensive. As a result, RND would require more work and consequently more CPU time per iteration as opposed to LEW to reconstruct the range space direction, p_y and the multipliers for equality constraints, *v.* This matrix can be simplified using the *Householder* transformation:

$$(I + aa^7 Y^1 = [I - a (I + a^T a)''' a^T] \quad (26)$$

Now the matrix to be inverted is only of the order of the number of degrees of freedom in the problem, and represents a very significant reduction in terms of number of computations required, as can be seen from the computational results shown in Section 6. With this transformation, and using the definition of *J3*> the expressions for p_y and *v* from eqn. 22 can be written as

$$p_y = - [i - a (I + a^7 a Y^X a^7] fi .$$

$$/.a. \qquad p^\wedge = - [fi - a (J + a^7 a Y^1 (a^7 >S)] , \quad (27)$$

and similarly,

$$v = - (V_y h Y^1 [I - a (I + a^T a Y^X a^7] Y^7 (V^> \cdot Vtf u) \quad (28)$$

With this simplification, we are now able to state the RND algorithm:

- Choose a convergence tolerance TOL.
- Choose a starting point z_0 and initialize the approximation to the null space projection of the Hessian of the Lagrangian, $(Z^7 B Z)_Q$, to Identity. Set the iteration counter k to 0.

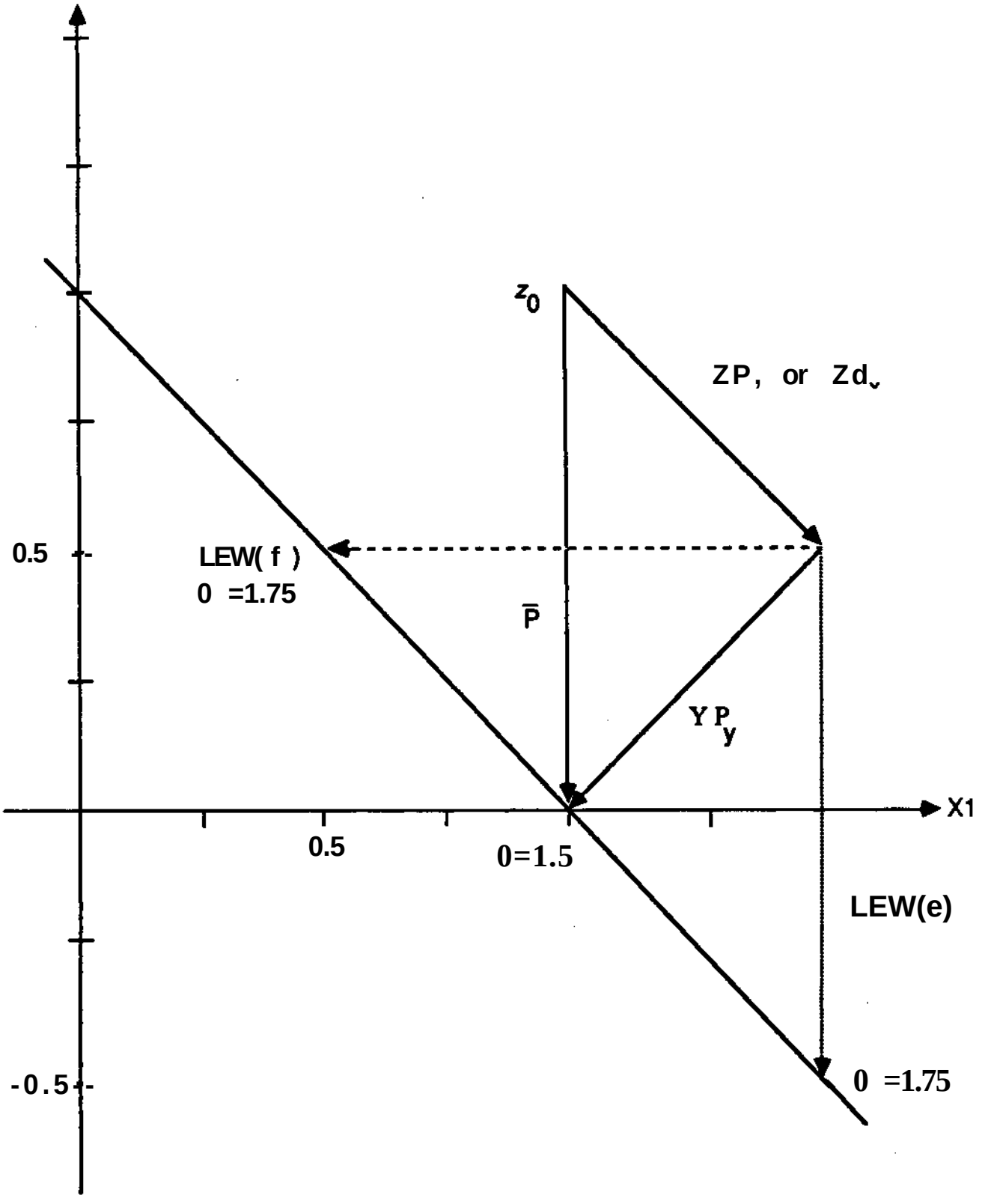

Figure 4-1: Geometric representation of *Example* problem

- **At each iteration k:**
  - **Evaluate <*p*, VV>, /?, V/?, *g* and ∇g**
  - **Set up a system of linear equations for L/U decomposition as an augmented matrix (say)**

$$A = [V/\ i\ -V/\ i\ -/,]$$

  - **Calculate using L/U decomposition * and *J3* from A Set up another augmented matrix (say)**

$$C = [(I + *^{T}*)\ !\ (*V)]$$

From C, compute p_y by L/U decomposition and matrix multiplication, (eqn. 27)

  - **Solve the QP to obtain p_z and *u.* (eqn. 21)**
  - **Estimate the multipliers for the equalities *v* for line search, (eqn. 28)**
  - **If the error in the first order necessary conditions (eqn. 1) is less than TOL then stop. Else**
  - **Perform line search to get X, the step length along *p.***
  - **Set $z_{k+1} = z_k + X(Yp_y + Zp_z)$**
  - **Update the projected Hessian $(Z^T B Z)$ using BFGS update rule with Powell's (1977) positive-definite correction.**
  - **Set k = k + 1 and repeat:**

5.1. Desirable Features Of Implementation Of RND Algorithm

- **The implementation of the algorithm is greatly facilitated by the use of a robust and flexible L/U decomposition routine which employs partial pivoting to put the maximum valued elements along the diagonal. The routine has two options which are controlled by setting a parameter to 0 or 1. If the parameter is 0 then the routine solves a system of form Ax = b, and stores the order of pivoting in an array. If on the second call to the routine, the parameter is set to 1, then the routine solves for $A^T x = b$, using the L and U factors estimated during the first call, thereby reducing the number of computations drastically, as there is no need to refactor the coefficient matrix. This was employed in calculating the L/U factors of the matrix $(V_y h)^1$ for calculating p_y and then the factors of $(V_y h)$ for the multiplier estimate.**

The same is true for the factorization of the matrix $(I + a^T a)$, which is symmetric.

- Dynamic storage allocation - the program contains at most a few arrays that must be dimensioned by the user, depending on the problem size. All the data arrays are stored in these. This places a very high upper limit on the size of the problems that can be accommodated by this method.

- The decomposition algorithm has been implemented as an easy to use part of a larger system which handles the solution of the QP, the line search and the BFGS update. A reliable QP solver, the QPSOL (Gill et al. (1983)) FORTRAN package for Quadratic programming, an efficient augmented Lagrangian-based line search strategy to guarantee global convergence (Biegler and Cuthrell (1985)) and a provision for automatic variable and constraint scaling all contribute toward a fast and robust package.

- The degree of decomposition is user controlled; one can specify the number of equality/active inequality constraints to be eliminated. In addition there is a run time capability that allows the user to specify an arbitrary partition of the variables into dependents or pivoted and independents.

- A very important feature of the implementation is the ability to handle singular Jacobians. The L/U routine, if the coefficient matrix is singular for a particular choice of basis, indicates the location of the singular element by specifying the row and column numbers. Based on this information, the program, after reconstructing the coefficient matrix, pivots in the maximum valued element by scanning along the singular row into the singular element location and altering the partition indicators for the variables accordingly. Although this option is adequate for the choice of dependents and independents at the onset of the problem, caution should be exercised if the Jacobian becomes singular at a subsequent stage, since the effect of a change in basis on the second order information, accumulated so far in the form of a BFGS update for a particular partitioning, should be considered. One obvious option is to reset the projected Hessian approximation, $(Z^T B Z)$, to Identity and restart with the new basis. This was found by experience to be safe and effective, serving only to slightly slow down the convergence of this method. A second option is to repartition the variables but continue based on the second order approximation obtained before. Although this seems to work on some of the problems we tested, it is not recommended. The third and probably the most reliable option theoretically, is to compensate for the change in basis, by considering explicitly the influence of the basis Z for the null space on the update. This requires an explicit form for the transformation matrix M which permits the orthogonal basis to be expressed as a linear combination of the columns of an orthonormal basis, *i.e.*,

$$Z = \bar{Z} M \tag{29}$$

where the symbol with an overbar refers to the orthonormal basis. It can be seen that the columns of the matrix (say) E, given by

$$E = Z\,(Z^T Z)^{-1/2} = Z\,(M)^{-1} = \bar{Z} \tag{30}$$

form an orthonormal basis for the null space of ∇h^T, since each column of E is just a linear combination of the columns of Z and by direct calculation we see that $E^T E$ is Identity. This would however require the storage of the basis Z for the previous iteration too and a *Cholesky* factorization to compute E. Although we did not implement this approach, we mention this option here for completeness.

6. Computational Results

The algorithm described has been tested on a wide variety of test problems against an implementation of the Locke, Edahl and Westerberg algorithm (LEW) and the full SQP method. The programs were written in FORTRAN 77 and run on a DEC-20 computer at the Carnegie-Mellon University. Double precision computation was used throughout. Comparisons are based on the number of iterations *i.e.* the number of QP's that needed to be solved. The results are presented in tables 1, 2 and 3 which also indicate whether the algorithms solved the problems to the desired tolerance or terminated with a line search failure away from the optimum. All the results reported are for unscaled runs and the CPU times required, in seconds, to solve the problems are also given.

The test problems can be classified into three groups. The first example problem can be classified as a separate class in itself, as it is a demonstration of the independence of the RND method to the initial basis choice and how the RND method parallels the progress of the full SQP method based on the initial approximation to the projected Hessian, $(Z^T B\ Z)$. This problem is a limiting case, with linear equality constraints and with a feasible starting point, at which point the RND and LEW methods are identical. The next set of problems are those employed by Nocedal and Overton. These problems are excellent for comparative testing and convergence analysis, as all these problems were initialized sufficiently close to the optimum, ensuring convergence. Thus, these are true indicators of the local convergence of the methods tested and excellent yardsticks for conclusions on performance. The third class is a myriad of test problems from diverse sources with known solutions. The purpose of this set is to compare performance of these algorithms when initialized away from the optimum and the effect of change of initial basis (variable partition). All the problems above were run with at least two

different and arbitrarily selected choices of dependent and independent variables, which are indicated in the discussions of individual problems. These were excellent indicators of the sensitivity of the algorithms to the manner of partitioning. Further discussion of individual problems, as warranted, will be done after stating the problems solved.

PROBLEM INDEX	SPECS. N.M.MEQ	FULL SQP	RND or LEW $(Z^T S Z)_0 = Z^r Z$	RND or LEW $\langle Z^T S 2)_0 = I$
l[a]	5,3,3	16 U.4E-15K2.93]	16 (1.4E-15K3.15]	20 (6.8E-16K3.89]
ltb]			16 (1.4E-15)	21 (5.2E-18)
Uc]			16 (1.4E-15)	17 (1.1E-15)

Table 6-1: Constant Null and Range space - *Linear* problem

The first test problem solved is problem number 50 from Hock and Schittkowski (1980). The main reasons for selecting this problem is that all the constraints are linear equalities and the initial starting point is feasible. The variables in the problem are unbounded above and below, which gives a problem without inequalities. Thus, the bases for the null and range space viz. Z and *Y* are essentially constants and given the initial feasible point, all subsequent iterates will be chosen to be feasible and hence the motion in the range space p_y will be essentially zero for all iterates, the movement being only in p_z in the null space. In fact, for the linearly constrained case with a non-feasible starting point, solution of the range space equations at the first iteration should yield a feasible point and p_y will be zero thereafter. One expect superlinear convergence for the RND algorithm under such circumstances.

The results are given in table 6-1, where the letters used in reference to each problem indicate the number of different partitions of the variables into dependents and independents that were considered. In this context, [a] refers to the original problem as is, with the variables 3, 4 and 5 chosen as dependent. For options [b] and [c], the pivoted variables are 1, 2 and 5 and 1,2 and 3 respectively. The specifications for these problems are given under the column SPECS., where, N refers to the number of variables, M to the number of constraints and MEQ to the number of equality constraints. The first number in each of the

other three columns, corresponding to the algorithms, refers to the number of iterations required and the second number in parentheses, the final error in the first order necessary conditions, which will be referred to as the Kuhn-Tucker error (KTE). In addition, a third number is reported for the base case option ([a]), which is the CPU time in seconds required to solve the problem. The times for the other options are comparable, and hence, not reported explicitly.

The full SQP method was solved with B initialized to Identity and took 16 iterations to solve the problem, requiring on average 0.183 CPU seconds per iteration. In all the test problems solved, the null space projection of the Hessian viz. $(Z^1 B\ Z)$ was initialized at Identity. Now, if we had chosen the bases to be orthonormal, this choice for RND would be identical to choice of Identity for the full SQP Hessian, since (Z Z) would be Identity from orthonormal property. Thus, the progress of the RND method with orthonormal bases and the full SQP method should be identical for the above problem, since the only progress of solution is due to movement in the null space. This is true, irrespective of which variables are picked to be dependent, since the progress of the RND method merely parallels that of the solution obtained from solving the undecomposed QP.

With our choice of nonorthonormal but orthogonal bases, and the two-sided projection of the Hessian set to Identity, we cannot expect the QP solution to duplicate the undecomposed QP solution, as the $(Z^T\ Z)$ term is no longer Identity. But, it is obvious that if we initialize instead, the null space projection as $(Z^T\ Z)$ rather than Identity, we take the same step as the full SQP method at the first iteration and hence, with absence of movement in the Range space, the progress of the RND method with orthogonal bases and the full SQP method are identical. This flexibility offered by the RND method in terms of initialization is exploited in this test problem, where, by appropriate initialization, irrespective of the set of dependent variables the RND method merely reduces to the full SQP method. This is also evident by re-examining the solution obtained from the *Example* problem.

The LEW method has been shown before to have the same null space movement as the RND method, with the difference being only in the step taken to reach the tangent subspace of the active constraints, which is an orthogonal move in the case of the RND method. In the absence of range space movement, as is the case with this problem, the same QP's will be solved by both decomposition strategies, and hence the two methods are identical.

As is evident (rom the results, with the projectedf Hessian initialized to Identity, the decomposition strategies take slightly different numbers of SQP iterations to converge to the optimum, depending on the choice of variable sets as dependent and independent Also evident, is the fact that with an appropriate initialization of the projected Hessian, (which does not require additional computation as the basis for the null space Z is calculated anyway) the RND and LEW method are insensitive to variable partitions, and duplicate the solution progress obtained by solving an undecomposed QP at each iteration. Thus one can make a definitive conclusion, that in a problem with only linear equality constraints and a feasible starting point, the RND and LEW methods are identical, and the sensitivity of these methods to variable sets is influenced only by the Hessian initialization. The decomposition strategies require on average 0.194 CPU seconds per iteration to solve this problem. A more significant result is in terms of the CPU time required to solve the QP at each iteration (not reported), where, the undecomposed SQP algorithm takes about 0.052 CPU seconds per iteration, while the decomposition strategies require 0.028 CPU seconds, a significant decrease.

For a problem with nonlinear constraints however, the manner in which the projected Hessian matrix, $(Z^T B Z)$, is initialized is generally arbitrary, and .a choice of Identity (I) matrix is as effective as any other choice. Under these circumstances, the two decomposition strategies are significantly different because of their range space movements, which is the source of LEWs inconsistency. One would expect the RND method because of its orthogonal moves to the constraint surface, to be less sensitive to choices of dependent and independent variables, a factor which is borne out in the results of the rest of the problems solved to test these algorithms.

Details and the provenance of the problems solved in table 6-2, can be found in Nocedal and Overton (1985). All problems were solved with a value for TOL of 10^{-10} and the projected Hessian approximation was initialized to Identity. Nocedal and Overton, however, initialized the null space projection of the Hessian by differencing the gradient of the Lagrangian using forward differences at Z_k along the columns of the null space basis $\bar{Z}$ and solved the problems to a tolerance of 10^{-8} only. This explains the slightly fewer iterations required by them to solve these problems using orthonormal bases. However, the technique of computing finite-difference along the columns of $\bar{Z}$, which is very successful for small problems, is too expensive in the large-scale case, because of the effort required to form $\bar{Z}$. In addition, even if the current

PROBLEM INDEX	SPECS. N,M=MEQ	FULL SQP	RND	LEW
2[a]	2,1	6 (7.2E-12H0.81]	7 (2.7E-13K1.23]	7 (6.2E-11K1.11]
2tb]			7 (7.4E-11)	9 (8.0E-14)
3Ca]	3,2	7 (3.3E-15K1.01]	7 (1.5E-13)[1.21]	10 (1.2E-11K1.58]
3[b]			7 (1.1E-13)	10 (1.2E-16)
3[c]			10 (7.7E-11)	>50+
4[a]	5,3	8 (1.5E-10)[1.49]	9 (2.0E-12)[1.81]	11 (1.2E-13)t2.14]
4[b]			9 (9.0E-13)	13 (9.0E-13)
4[c]			11 (2.7E-13)	>50+
5ta]	5,3	7 (2.0E-12K1.36]	11 (4.7E-11H2.16]	12 (1.8E-13)[2.23]
5[b]			11 (9.5E-12)	11 (1.0E-12)
5[c]			10 (1.9E-12)	36 (6.7E-14)
6[a]	7,2	22 (1.3E-11K6.11]	17 (2.1E-12)[4.52]	17 (3.3E-13)[4.48]
6[b]			19 (1.1E-10)	26 (3.1E-10)
6tc]			20 (1.3E-11)	9*
7[a]	10,3	39 (4.9E-10K 17.75]	33 (5.9E-10)[13.54]	40 (9.4E-10)[15.76]
7[b]			34 (7.0E-12)	52 (2.0E-10)
7[c]			34 (7.3E-10)	50 (9.1E-10)
8[a]	8,4	12 (6.4E-10)[3.59]	15 (2.8E-11H4.12]	13 (1.7E-11)[3.34]
8[b]			15 (1.4E-10)	11 (5.3E-10)
8[c]			13 (3.2E-12)	35 (8.3E-13)

* Terminates at an infeasible point

t Terminated owing to serious ill-conditioning at infeasible point

Table 6-2: Local Convergence tests of SQP algorithms - *Nonlinear* problems

approximation of the Hessian of the Lagrangian function is available, it is probably too

expensive to form its null space projection (Gill et al. (1981)). All the constraints were specified as equalities, and for the last three problems, we followed Nocedal and Overton's approach of specifying only the active constraints. As in table 6-1, the final KTE values for all cases and the CPU times for the base cases are also reported.

The convention was to partition the variables set into decisions first and dependents next. So, in all the base cases ([a]), the variables towards the end of the variable set, equal to the number of equality constraints were chosen as the pivoted variables, initially. For example, in problem 4 which has 5 variables and 3 equality constraints, for the [a] option, variables 3,4 and 5 were chosen to be in the initial basis set Of course, if the Jacobian becomes singular during the course of optimization, an alternate, appropriate basis would be chosen, different from the initial set specified. When this happened, we picked the initial basis sets for the other options ([b] and [c]) in such a way as to avoid overlaps, with either the initial sets specified before or alternate sets picked thereof, owing to singular Jacobians. The variables picked as pivoted initially for option [b] are variable 1 for problem 2; 1 and 3 for problem 3; 1, 3 and 4 for problem 4; 1, 4 and 5 for problem 5; 1 and 2 for problem 6; 1, 8 and 9 for problem 7; and 1, 5, 6 and 8 for problem number 8. For case [c], the initial basis sets were variables 1 and 2 for problem 3; 1, 2 and 3 for problem 4; 1, 2 and 5 for problem 5; 2 and 3 for problem 6; 1 , 5 and 9 for problem 7; and finally variables 1, 2, 5 and 8 for problem 8.

As is evidenced by the results, RND is generally faster than LEW, although nothing in the theoretical development always guarantees this. RND, with the exception of cases 8[a] and 8[b], never requires more QP solutions than LEW. But the most significant conclusion that can be drawn from these results, is in terms of the sensitivity of the methods to the manner of variable partitioning, even when initialized close to the optimum. While the RND method is almost insensitive to this, and requires almost the same number of QP solutions irrespective of the choice of the initial basis, the LEW algorithm is highly dependent on this. Not only do the number of iterations required by LEW vary considerably, but in a couple of cases (3[c] and 4[c]), the algorithm was terminated after more than 50 iterations at points radically away from the optimum, although it was initialized close to it In addition, in one other instance 6[c], LEW terminates after only 9 iterations due to line search failures owing to ill-conditioning, at a highly infeasible point

The times reported, when averaged per iteration, show that RND is only slightly slower than LEW, and that the additional work required in RND to reconstruct the multipliers for the equality constraints is not computationally significant Further, as the results suggest, the RND method never requires significantly more iterations, and in some instances, fewer QP solutions, than the undecomposed SQP algorithm.

The results given in table 6-3, are for randomly selected test problems, where starting points suggested by their authors were used. The principal purpose of solving these test problems is to rate the performance and robustness of the algorithms in solving problems initialized away from the solution. Further, in each instance, at least two different choices of initial basis were tried out to gauge the sensitivity of the algorithms to this choice. A suitable value for TOL was picked for each problem based on the magnitude of the objective function to obtain sensible results and the approximation to the null space projection of the Hessian was initialized at Identity for all algorithms. Unless otherwise indicated in table 6-3, all solutions satisfied the chosen value of TOL. The other details are as before in table 6-1.

Here the problems numbered from 9 to 13 are from Himmelblau (1972), and the number in parentheses is the problem number in the reference. Problem number 14 is described in detail in Locke, Edahl and Westerberg (1983) and is a simple optimization problem with 4 variables and 2 equality constraints. Problem 15 is the alkylation process model of Bracken and McCormick (1968) which was solved by Berna, Locke and Westerberg (1980). The last problem 16, is the chemical process optimization problem of Williams and Otto. In reference to this problem, the numbers used in the problem column indicate the three different starting points (see, Ray and Szekely (1973)) that were used.

For the first 5 problems, the variable partitions for the base cases ([a]) were done in the same manner as outlined before. For the [b] option, the dependent variable sets are variable 2 for problem number 9, variables 8, 9 and 10 for problem 11, variables 7, 9 and 10 for problem 12 and variable 1 for problem 13. For the [c] option, the choices are variables 2, 3 and 7 for problem 12. As the results of these problems indicate, RND never requires more iterations than LEW. Further, while RND solves all these problems, irrespective of the manner of partitioning of the variables, to optimum, the LEW algorithm fails to solve one problem 12 [4a] completely, terminating very far from the optimum, but is able to solve the same

PROBLEM INDEX	SPECS. N.M.MEQ	FULL SQP	RND	LEW
9[la]	2,2,1	6 (2.8E-12K0.87]	6 (4.5E-07K0.92]	7 (2.5E-09K1.01]
9[lb]			6 (6.0E-06)	6 (6.0E-06)
10C3]	2,3,0	6 (4.1E-10K0.82]	6 (4.1E-10)[0.82]	6 (4.1E-10K0.79]
ll[4a]	10,3,3	18 (5.0E-09K7.52]	32 (9.6E-06)[12.81]	37 (4.1E-07)[13.98]
llC4b]			32 (6.0E-07)	38* (2.0E-04)
12[4a]	10,3,3	51 (8.0E-09K 25.30]	57 (3.8E-08K 26.02]	9*
12[4b]			56 (4.5E-07)	94 (4.4E-06K50.02]
12[4c]			54 (4.0E-08)	59 (1.4E-07)
13t5a]	3,2,1	9 (5.0E-13)[1.23]	10 (7.2E-09)[1.59]	11 (1.0E-10)[1.73]
13t5b]			11 (2.4E-11)	16* (2.1E-06)
14[a]	4,2,2	7 (6.0E-11K1.11]	11 (1.4E-10)[1.83]	10 (8.0E-13)[1.66]
14[b]			17 (7.3E-10)	> 50*
14[c]			7 (4.2E-11)	10*
15[a]	10,11,3	31 (2.4E-07K13.13]	29 (7.1E-07)[11.57]	31^{+} (2.1E-05K11.17]
15[b]			27 (1.1E-07)	33 (1.7E-06)
16[la]	10,7,5	196 (2.0E-11K80.36]	89 (5.3E-07)[30.36]	103 (5.2E-07)[34.69]
16[lb]			38* (6.0E-04)	40^{+} (1.2E-05)
16[2a]		122*	96 (9.3E-06)	113 (8.4E-09)
16[2b]			161 (8.0E-06)	186^{+} (9.2E-04)
16C3a]		79*	147 (2.5E-10)	149 (7.8E-10)
16E3b]			50 (7.0E-06)	59* (4.0E-04)

* Terminates at an infeasible point + TOL not satisfied

• Terminates at a significantly different local minima

Table 6-3: Comparison of performance on general test problems

problem for alternate initial basis choices, requiring highly varying number of iterations. In

two other instances, LEW fails to satisfy the TOL specified, though the final value of the objective function is correct, terminating because of line search failures. As before, while the results for the RND method indicate independence of the choice of dependents and independents, the results for the LEW method indicate a strong dependence on this choice. The only purpose of solving problem 10, is to show that in the absence of equality constraints, these decomposition strategies reduce to the full SQP method.

Significant results were obtained during the solution of the next problem 14, which was solved by Locke et al. (1983). Details of this problem are also given in the same reference. The authors report an optimum value of -3.305 for this problem. For the choice of dependent or pivoted variables which the authors used when solving the problem, which are variables 3 and 4, RND requires one more SQP iteration than LEW. This is easily explained by analyzing the iterates for RND, where the search direction given by the QP at the first iteration leads to a singular Jacobian, warranting the choice of an alternate set of pivoted variables viz. variables 1 and 4, thereby slowing down the performance slightly. The same is true for the second case 14[b] for RND, where variables 1 and 3 were picked as pivoted variables. LEW, for this initial set becomes highly ill-conditioned and had to be terminated after 50 iterations at an infeasible point. Since, in both these cases the variables 1 and 4 constitute the final pivot set, they were chosen as the initial basis for option 14[c], where, RND requires only 7 iterations as the Jacobian remains non-singular throughout. However, for this choice LEW terminates with line search failures at a value of -3.30501 for the objective function and fails to converge the equality constraints to the required tolerance. The same is true of LEW for most other possible combinations of dependent and independent variable, thus emphasizing the marked inconsistency in the performance of this algorithm.

Problem 15 employs the original formulation of the alkylation process optimization of Bracken and McCormick (1968) and is schematically represented in fig. 6-1. It is a simplified model of an alkylation process, where fresh olefins and isobutane are added to a reactor. Sulfuric acid is added to the reactor as a catalyst and spent acid is removed. The effluent from the reactor is sent to a fractionator, where the alkylate product is separated from the unreacted isobutane, which is recycled to the reactor. Table 6-4 gives the mathematical formulation employed and table 6-5 gives the details of the bounds employed on the variables as well as their starting and optimal values. The problem has 10 variables. Excluding the bounds on the variables, is also has 11 constraints, of which 3 are equalities.

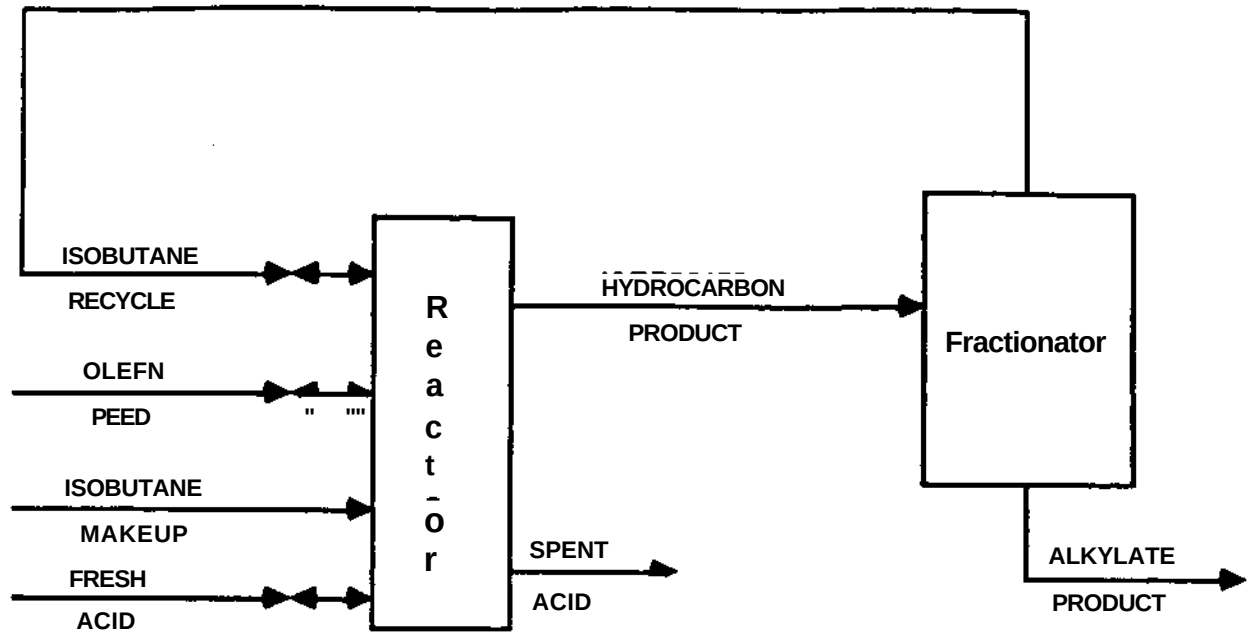

Figure 6-1: Bracken and McCormick alkylation process

The results for the two decomposition algorithms are almost identical (although LEW fails to satisfy the tolerance for the first option, terminating with line search failures) and are better than the values reported by Berna et al. (1980). Here, the pivoted variables chosen initially for problem 15[b] are 4, 5 and 6.

Although the results presented here are for the unsealed problem, the automatic variable and constraint scaling procedure proposed by Biegler and Cuthrell (1985) has also been implemented in the RND algorithm. Since the magnitude of the variables in the alkylation problem vary considerably, we decided to solve the problem using this scaling procedure. The scaled run only took 13 iterations to solve the problem for the RND method for both variable sets, a reduction of about 15 iterations. These results are very encouraging and indicate that the scaling procedure will be extremely beneficial for solving much larger problems.

The last example in this class is the Williams and Otto flowsheeting problem. The process here has many of the attributes typical of a chemical process plant, involving common unit operations like separation (through decanting and distillation) and reaction, and has topological features like recycles, bleeds, feed and product streams, characteristic of a flowsheet The

$$Min\ -<p = 5.04x1 + 0.035x2 + 10.0x3 + 3.36x5 - 0.063x4x7$$

s.L $x2 + x5 - xlx8 = 0$

$9.8E4X3 - (x4x9 + 1.0E3x3)x6 = 0$

$1.22x4 - xl - x5 = 0$

$-xKl.12 + 0.13167x8 - 0.00667x8^2) + 0.99x4 £ 0$

$xl(1.12 + 0.13167x8 - 0.00667x8^2) - X4/0.99 £ 0$

$-86.35 - 1.098x8 + 0.038x8^2 - 0.325(x6 - 89.0) + 0.99x7 £ 0$

$86.35 + 1.098x8 - 0.038x8^2 + 0.325(x6 - 89.0) - X7/0.99 £ 0$

$-35.82 + 0.222x10 + 0.9x9 £ 0$

$35.82 - 0.222x10 - X9/0.9 <, 0$

$133.0 - 3.0x7 + 0.99x10 \leq 0$

$-133.0 + 3.0x7 - xlO/0.99 \leq 0$

Table 6-4: Formulation for Alkylation Optimization Process

objective of the process is to manufacture a certain product P from two chemicals A and B, and the three reactions involved are as follows:

$$A + B \text{ -» } C$$

$$C + B \text{ -> } P + E$$

$$P + C \text{ •* } W$$

The plant, shown in fig. 6-2, produces in addition to the desired product P, a heavy oily waste product W, which has to be disposed. Components C and E are intermediate by-products of no commercial value as independent commodities and are used as plant fuel. The effluent from the reactor is cooled in a heat exchanger and the waste W is removed with a decanter. Subsequently product P is removed in the overhead of a distillation column. Owing to thermodynamic restrictions, a fraction of the product is retained in the bottoms of the column. The bottom product is then split into two streams; one is recycled to the reactor and the other is used as plant fuel.

Variable	Lower Bound	Upper Bound	Starting value	Optimum value
xl	0	2000	1745	1698.09
x2	0	16000	12000	15818.2
x3	0	120	110	54.1041
x4	0	5000	3048	3031.22
x5	0	2000	1974	2000.0
x6	85	93	89.2	90.1156
x7	90	95	92.8	95.0
x8	3	12	8.0	10.4931
x9	1.2	4	3.6	1.56164
xlO	145	162	145	153.535
<p	—	—	872.387	1768.81

Table 6-5: Summary of Bracken and McCormick problem

The modular nature of the flowsheet enables the entire process to be represented in terms of equations for each unit, whereby, the output flows are calculated from the inputs and equipment parameters which are specified. Employing the same notation as in fig. 6-2 for mass flowrates of the streams in the flowsheet and using the convention that superscripts represent individual component flows, the calculation sequence is given below. This formulation is similar to one presented in Ostrovsky and Berezhinsky (1984).

1. Stirred-Tank Reactor Equations:

$$F_R = F_A + F_B + R^A + R^B + R^C + R^E + R^P$$

$$F_R^A = F_A + R^A - K_1 (F_R^A F_R^B) \, Vp/F_R^2$$

$$F_R^B = F_B + R^B + \left[- K_1 (F_R^A F_R^B) - K_2 (F_R^B F_R^C) \right] V\rho/F_R^2$$

$$F_R^C = R^C \cdot [-<M_c / M_\pounds) \, K_2 \, (F_R^B \, F_R^C) - K_3 \, (F_R^C \, F_R^P)$$

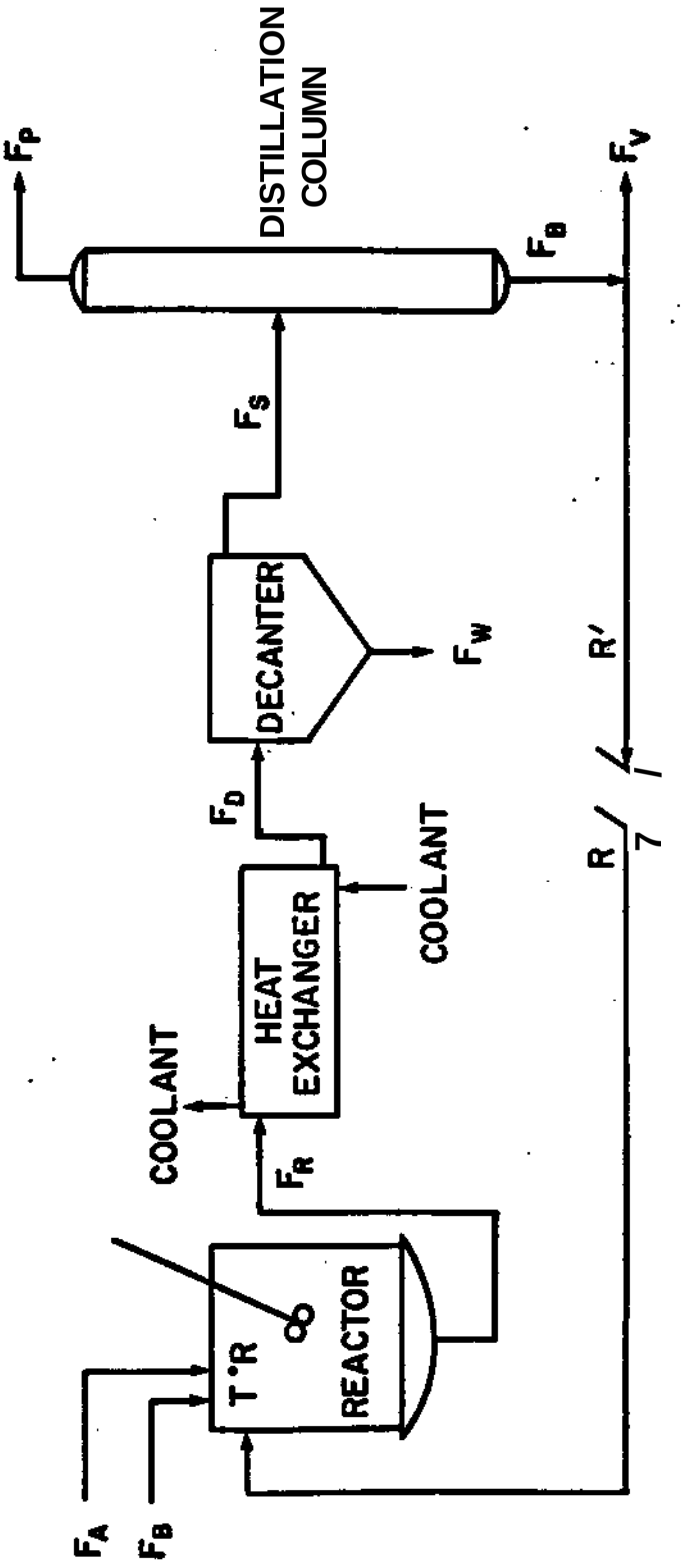

Figure [illegible] t of [illegible] t - Williams S O O S

$$\bullet \langle M_C / M_{\pounds}) K_1 (F_R^A F_R^B)] \; V\rho / F_R^2$$

$$F_R^P = R^c - \langle M_p / M_c) K_3 (F_R^P F_R^C) \; V\rho / F_R^2$$

$$\bullet (M_p / M_B) K_2 (F_R^C F_R^{\beta})$$

$$F_R^E = R^E + (M_E / M_B) K_2 (F_R^B F_R^C) \; V\rho / F_R^2$$

$$F_R^W = (M_w / M_y) C_3 K_R (F_R^P F_R^C) \; V/>/F_R^2$$

$$K_i = A_i \exp (-E_i /RT) \qquad i = 1,2,3$$

2. Heat Exchanger Equations: (No energy balances; trivial mass balances)

$$F_D^i = F_R^j, \qquad i = A,B,C,E,P,W$$

3. Decanter Equation: (Perfect separation of waste assumed)

$$F_W^{\bullet W} = F_D^W, \qquad F_S^W = 0 \qquad i = A,B,C,E,P$$

$$F_W^! = F_D^!, \qquad F_S^! = 0, \qquad i =$$

4. Distillation Column Equations:

$$F_P^P = F_S^P - 0.1 F_S^E, \qquad V = 0.1 F_S^E$$

$$F_B^i = F_S^i, \qquad F_P^i = 0, \qquad i = A.B.CE$$

5. Flow Splitter Equations:

$$F_V^i = \eta F_D^i$$

$$R^i = (1 - \eta) F_B^i \qquad i = A.B,C.E,P$$

6. Recycle or Tear Equations:

$$R^{\#} * - R^* = 0, \qquad i = A,B,CE,P$$

The objective function here is the annual rate of return on fixed investment and given in terms of the net sales minus the fixed charge, raw material, utility and waste disposal costs. It can be expressed as follows:

$$\varphi = [C F + C F - C F - C F - C F - C F + C \; v / l / (C \; V/>)$$

Details regarding the kinetic and process parameters as well as the cost coefficients can be found in numerous literature references where this problem has* been solved (see, e.g., DiBella and Stevens (1965) and Ray and Szekely (1973)). In addition to the tear equations, additional constraints on the problem are in the form of inequalities that bound the production rate and variables as follows:

$$580°R \leq T \leq 680°R$$

$$0 \leq F_p \leq 4763$$

$$30 \leq V \leq 100$$

$$0 < {}_n < 0.99$$

All flows ≥ 0

which gives a final formulation with 10 variables and excluding the bounds on variables, 7 constraints, of which 5 are equality tear constraints. Thus for a fixed feed and density />, a choice of any five variables can be used as decisions for optimization.

The 3 different starting points employed for the variables and the initial and final values of objective function are given in table 6-6. In addition, the final variable values obtained for the third starting set with the RND option are also given. Without scaling, the same optimum value is found from the different starting points, so that a return of 131.423% is considered optimal for this problem.

This problem is however plagued with a myriad of local optima and very exacting tolerances had to be specified before the optimal solution could be obtained. In fact, in the couple of instances where the full SQP method failed, we found that the constraints were all converged and the KTE was as low as 10^{-9}. Interestingly however, both decomposition schemes progressed to the solutions without line search failures, and with acceptable tolerances. The problem with local minima was accentuated when we changed the choice of pivoted variables to variables 1, 2, 3 *A* and 5 for the [b] options, with both decomposition schemes terminating at local minima. Examination of these results indicated that all the variables were at their lower bounds, which is zero for the flowrates and the final value of the objective function was close to zero. Here, we resolved the problem by placing reasonable lower bounds on the variables, and obtained the results given in table 6-3. Since this problem contains local optima

Variable	Starting set 1	Starting set 2	Starting set 3	Optimum values (3)
F_s^A	8,820	6,000	18,187	46,261.4
K	39,910	35,000	60,815	143,281
F_s^C	2,360	10,000	3,331	7,585.83
F_s^P	7,890	6,000	10,817	18,826.5
K	31,660	15,000	60,542	141,847
F_A	11,540	10,000	13,546	13,164.3
F_B	31,230	40,000	- 31,523	29,991.2
V	60	60	60	30
T	610	.610	656	674.36
V	0.5806	0.714	0.7536	0.8998
initial<*P(%)*	124.624	1.2436	81.9102	—
final <*p(%)*	131.423	131.423	131.423	—

Table 6-6: Summary of Williams and Otto problem

and extreme nonlinearities, we observe that this example may not be suitable for making conclusions regarding the sensitivity of the RND and LEW algorithms toward the choice of dependent variables. However, as can be seen from the results, the RND method outperforms the LEW algorithm in each case.

7. Conclusions

In this paper, we have developed and implemented a new Range and Null space decomposition strategy (RND) based on nonorthonormal projections. RND uses orthogonal bases (but not orthonormal columns) for the Range and Null spaces of the matrix of the active constraint gradients. These bases are sparse formulations based on a partitioning of the variable set They use the gradient information that is already available and do not require explicit computation or storage. Hence, RND is ideally suited for implementations where the Jacobian is large and sparse.

RND, has been shown to have the same convergence characteristics as the full SQP method, but requires only the computation of the null-space projection of the Hessian matrix, which is of the dimension of the number of degrees of freedom in the problem. This strategy is implemented in a robust manner and compensates for difficulties like singular Jacobians. It thus overcomes the problems encountered in the implementation of previous decomposition strategies proposed by Westerberg et al.

Numerical comparison derived from the solution of a number of highly varied nonlinear and process optimization problems demonstrates the effectiveness of this strategy, compared to other SQP methods. In particular, they show that it avoids the inconsistent behavior found with the LEW method and requires about the same number of iterations as the full SQP method, but by solving much smaller quadratic programs.

RND can also be extended to include specific decomposition strategies to factor sparse gradient matrices where the non-zero elements occur in a specific pattern. Finally, the method can also be extended to exploit the fact that only a part of the Jacobian changes from one iteration to another, since the rows of the constraint gradient matrix that correspond to linear constraints and simple bounds remain constant. Here, refactoring of that part of the basis to compute the inverse (implicitly) can be avoided, as it is possible to recur the factorization from iteration to iteration using the partitioned inverse technique.

References

Berna, T. J., Locke, M. H. and Westerberg, A. W., "A New Approach to Optimization of Chemical Processes", *A/CHE J,* 26, 1, p. 37 (1980).

Biegler, L. T., "Improved Infeasible Path Optimization for Sequential Modular Simulators - I", *Comp. and Chem. Eng.,* 9, 3, p. 245 (1985).

Biegler, L. T. and Cuthrell, J. E., "Improved Infeasible Path Optimization for Sequential Modular Simulators, Part II: The Optimization Algorithm", *Comp. and Chem. Eng.,* 9, 3, p. 257 (1985).

Boggs, P. T., Tolle, J. W. and Wang, P., "On the Local Convergence of Quasi-Newton Method for Constrained Optimization", *SIAM J. Control Optim.,* 20, 161, (1982).

Bracken, J. and McCormick, G. P., *Selected Application of Nonlinear Programming,* John Wiley & ,Sons, Inc., New York (1968).

Chan, W. K. and Prince, R. G. H., "Application of the Chain Rule of Differentiation to Sequential Modular Flowsheet Optimization", *Comp. and Chem. Eng.,* 10, 3, p. 223-240 (1986).

Chen, H-S and Stadtherr, M. A., "A Simultaneous Modular Approach to Process Flowsheeting and Optimization", Parts I and II *A/CHE J,* 31, 11, p. 1843, (1985).

DiBella, C. W. and Stevens, W. F. *Process Optimization by Nonlinear Programming,* I & EC Proc. Des. Dev., 15, 1, p. 206 (1965).

Gill, P. E., Murray, W., Saunders, M. A. and Wright, M. H., "QP- Based Methods for Urge-Scale Nonlinearly Constrained Optimization", *Nonlinear Programming* 3 (1981).

Gill, P. E., Murray, W., Saunders, M. A. and Wright, M. H., "User's Guide for SOL/QPSOL: A Fortran Package for Quadratic Programming", Technical Report SOL 83-7 (1983).

Han, S.-P, "A Globally Convergent Method for Nonlinear Programming", *J. Opt. Theory and ConU* 22, 3, p. 297 (1977).

Himmelblau, D. M., *Applied Nonlinear Programming.* McGraw-Hill Book Co., New York (1972).

Hock, W. and Schittkowski, K., *Test Examples for Nonlinear Programming Codes,* Lecture Notes in Economics #187, Springer (1981).

Hutchison, H. P., Jackson, D. J. and Morton,W., 'The Development of an Equation-Oriented Flowsheet Simulation and Optimization Package-II. Examples and Results", *Comp. and Chem Eng.*, 10, 1, p. 31 (1986).

Jirapongphan, S., Boston, J. F., Britt, H. I. and L. B. Evans, "A Nonlinear Simultaneous Modular Algorithm for -Process Flowsheet Optimization", presented at 80th AIChE Meeting, Chicago (1980).

Kisala, T. P., ScD Thesis, Massachusetts Institute of Technology, Cambridge, MA (1985).

Lang, Y-D and Biegler.L. T., "A Unified Algorithm for Flowsheet Optimization", *Comp. and Chem. Eng.*, to appear (1986).

Locke, M. H., Edahl R. and Westerberg, A. W., "An Improved Successive Quadratic Programming Optimization Algorithm for Engineering Design Problems", *AICHE J*, 29, 5, (1983).

Luenberger, David G., *Linear and Nonlinear Programming*, Addison-Wesley Publishing Company, Massachusetts, 2nd Ed., (1984).

Nocedal, Jorge and Michael L. Overton, "Projected Hessian Updating Algorithms for Nonlinearly Constrained Optimization", *SIAM J. Numer. Anal.*, 22, 5, (1985).

Ostrovsky, G. M. and Berezhinsky, T. A., *Optimization of Chemical, Technological Processes: Theory and practice*, Khimia, Moscow (in Russian) (1984).

Powell, M. J. D., "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations", Dundee Conf. on Num Analysis (1977).

Powell, M. J. D., 'The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculations", *Nonlinear Programming* 3, O. Mangasarian, R. Meyer and S. Robinson, Eds., Academic Press, New York (1978a).

Ray, W. H. and Szekely, J., *Process Optimization*, Wiley & Sons, New York (1973).

Trevino-Lozano, R., ScD Thesis, Dept of Chemical Eng., MIT, Cambridge, Massachusetts (1985).