AN IMPROVED SUCCESSIVE QUADRATIC PROGRAMMING OPTIMIZATION ALGORITHM FOR ENGINEERING DESIGN PROBLEMS

by

M.H. Locke, R.H. Edahl & A.W. Westerberg

Decaraber, 1932

DRC-06-37-32

AN IMPROVED SUCCESSIVE QUADRATIC PROGRAMMING OPTIMIZATION ALGORITHM FOR ENGINEERING DESIGN PROBLEMS

by

Michael H. Locke*

Richard H. Edahl*

Arthur W. Westerbergt

June, 1982

tDepartment of Chemical Engineering

^School of Urban and Public Affairs

Carnegie-Mellon University

*Current address is Department of Chemical Engineering

University of Massachusetts, Amherst, Mass. 01003.

6

I

TABLE OF CONTENTS

1 ABSTRACT

A decomposition technique has been developed which greatly reduces the time and computer memory space required to implement the Han-Powell Optimization Algorithm. The technique requires the solution of a Quadratic Programming Problem (QPP) in the space of the decision variables only, rather than in the space of the entire set of variables as in the original algorithm published by Powell.

2 INTRODUCTION

Early attempts at optimal flowsheet simulation utilized a flowsheeting system in an "inner loop" to converge the equality constraints defining a flowsheet while an optimizer in the "outer loop" chose values of the parameters over which the flowsheet was optimized (see for example Friedman and Pinder, 1972). The optimizer generally used a pattern search method such as the Complex Method (Box, 1965) to choose values of the decision variables. Such methods are usually dependable, but require much computation and have trouble handling inequality constraints.

Recently, techniques have been developed which converge the equality constraints defining a flowsheet while moving towards the optimal values of the decision variables. The equality constraints may only be satisfied at the last iteration, at which time the decision variables reach their optimal values. One such technique is that developed by Powell (1977) based on work done by Han (1975). A quadratic program with linearized versions of the flowsheet equality and inequality constraints is set up and solved at each iteration. The quasi-Newton approximation to the Hessian Matrix for the QPP is the size of the number of variables in the problem, and is not generally sparse. A problem with 1000 variables would require 1,000,000 storage locations for the approximate Hessian Matrix.

Bema, Locke, and Westerberg (1980) developed an efficient implementation of Powell's algorithm to be used with a simulation program based on equation solving methods. That algorithm decomposes the computations for the problem so that the required Hessian is never handled directly. Biegler and Hughes (1981, 1982) have implemented the Han-Powell algorithm with a sequential modular simulation package. In their work the QPP is set up in the space of the decision variables and the torn variables of the recycle streams. Murtagh (1982) has developed a Lagrangian based optimization procedure (Robinson, 1974) which also simultaneously solves and optimizes flowsheet equations. The method uses the MINOS/AUGMENTED nonlinear programming system (Murtagh and Saunders, 1980) to solve and optimize the flowsheet equations. The technique seems quite promising when the flowsheet equations are mostly linear.

The technique presented here is a variation of Powell's algorithm (1977). As mentioned previously, this algorithm sets up a QPP in the space of the decision variables. It is computationally more efficient and conceptually cleaner than the algorithm described in Berna, Locke, and Westerberg (1980). The algorithm is very similar to an incorrect algorithm reported by Berna and two of the present authors several years ago (Berna, Locke, and Westerberg, 1978). Unfortunately that algorithm included an incorrect step, and it failed when tested on large problems. Work done by Edahl (1982) on the Han-Powell algorithm includes this algorithm as a subset of a broader class being investigated.

3 THEORY

The optimization problem can be stated as follows:

$$\min \Phi(z)$$

$$\text{s.t. } g(z) = 0$$

$$z_{min} \le z \le z_{max} \qquad \text{(P1)}$$

$$z \in E^{n+r}$$

$$g: E^{n+r} \to E^{n}$$

$$\Phi: E^{n+r} \to E^{1}$$

The Lagrangian of this problem is:

$$L(z,\tau,\kappa_{min},\kappa_{max}) = \Phi(z) - \tau^T g(z) - \kappa^T_{min}(z_{min} - z) + \kappa^T_{max}(z_{max} - z),$$

with τ the Lagrange multipliers on the equality constraints and κ_{min} and κ_{max} the Kuhn-Tucker multipliers on the lower and upper bounds of z, respectively.

The Kuhn-Tucker conditions for this problem are:

$$\kappa_{min}[z_{min} - z] = 0$$

$$*_{max} [z_{max} - z] = 0$$

$$«_{min}, «_{max} £ 0$$

For a typical flowsheet calculation of the type we are considering, n may be on the order of 10,000, with r on the order of 10. This formulation is quite general as general inequality constraints can be converted to equality constraints through the use of bounded slack variables.

Powell's approach to solving this problem is to linearize the equality constraints, assume a quadratic approximation to the objective function, and solve the resulting QPP. Variables are then updated by taking the step

$$Az = ad$$

where d is the step calculated by the QPP.

As stated previously, the disadvantage of using Powell's method is the size of the Hessian Matrix for the QPP. For a problem with 10,000 variables, the Hessian contains 100 million elements, far too many for even the most advanced machines to handle. By partitioning the variables into two sets, the independent or decision variables, u, and the dependent or pivoted variables, x, we now show how a QPP can be set up in the decision variables only.

Let x_k and u_R be the values of the dependent and independent variables respectively at the present iteration. Linearizing the equality constraints about this point gives:

$$g(x_{k+l},u_{k+1}) \sim g(x_k,u_k) + Og/3x^T)_k Ax_R + Og/3u^T)_k Au_k$$

Setting $9<^{x}{}_{k+r}{}^{u}{}_{k+1})$ equal to 0, and requiring that $0g/8x^T)_k$ be nonsingular yields:

$$Ax_R = - Og/3x^T)_k{}^{"1}\{g(x_k,u_k) \bullet Og/3u^T)_k Au_k>$$

For any choice of Au_{fc}, a value for *A^* can be calculated. The problem can then be stated as one to calculate Au_k*, the optimal change in the decision variables:

$$\min_{\Delta u_k} F(Au_k)$$

$$\text{s.t.} \quad u_{min} - u_k \leq \Delta u_k \leq u_{m.x} " u_k$$

$$x_{min} - x_k \le -(\partial g/\partial x^T)_k^{-1}\{g(u_k,x_k) + (\partial g/\partial u^T)_k \Delta u_k\} \le x_{max} - x_k \qquad \text{(P2)}$$

$$\Delta u_k \in E^r$$

$$x_k \in E^n$$

$$g: E^{n+r} \rightarrow E^n$$

where

$$F(\Delta u_k) = \Phi(x_k + \Delta x_k, u_k + \Delta u_k).$$

Expanding $F(\Delta u_k)$ in a Taylor series about the point $\Delta u_k = 0$ yields the following QPP associated with (P2):

$$\min_{\Delta u_k} \left\{ (Q(\Delta u_k) \,|\, Q(\Delta u_k) = a_k + b_k^T \Delta u_k + \tfrac{1}{2} \Delta u_k^T C_k \Delta u_k \right\}$$

$$\text{s.t.} \qquad u_{min} - u_k \le \Delta u_k \le u_{max} - u_k$$

$$-(\partial g/\partial x^T)_k^{-1}(\partial g/\partial u^T)_k \Delta u_k \le x_{max} - x_k + (\partial g/\partial x^T)_k^{-1} g(x_k,u_k)$$

$$(\partial g/\partial x^T)_k^{-1}(\partial g/\partial u^T)_k \Delta u_k \le x_k - x_{min} - (\partial g/\partial x^T)_k^{-1} g(x_k,u_k) \qquad \text{(P3)}$$

$$\Delta u_k \in E^r$$

$$x_k \in E^n$$

$$g: E^{n+r} \rightarrow E^n.$$

The vector b is the reduced gradient $\delta\Phi/\delta u$, calculated by doing a Taylor series expansion on the objective function:

$$\Phi(x+\Delta x,\ u+\Delta u) = \Phi(x,u) + (\partial\Phi/\partial x^T)\Delta x + (\partial\Phi/\partial u^T)\Delta u$$

but

$$Ax = - 0g/3x^{T}r^{1}\{g(x,u) \bullet @g/3u^{T})Au\}$$

then

$$A^{*} = - O^{*}/8x^{T})\{(3g/8x^{T})-^{1}\{g(x,u) \bullet Og/3u^{T})Au\}\} \bullet «4>/3u^{T})Au$$

$$= -»^{*}/8x^{T}X©g/ax^{T}r^{1}g<x.u) \bullet \{(34>/8u^{T}) - (34>/3x^{T}X(3g/3x^{T}r^{1}(3g/3u^{T})\}Au$$

The coefficient of Au gives the constrained derivative of • with respect to u:

$$b_{k} = 6^{*}1Su = O^{*}/du) - \{og/SxV'Og/Su^{1})\}^{7}©^{*}\wedge.$$

The fact that g(x,u) is not 0 does not effect the term b_R. It simply adds something to the scalar a_k.

The matrix C is an approximation to the Hessian of the Lagrange Function formed from the problem:

$$\min \$\{u,x(u))$$

$$\text{s.t.} \quad u_{mm} \cdot £ u £ u_{max}$$

$$x_{min} * x\{u) * x_{max} \qquad \text{(P4)}$$

$$x(u): \; g(x,u) = 0$$

$$u \; e \; E^{r}$$

$$x \; e \; E^{n}$$

The Lagrange of (P4) is:

$$L = \#\{u,x(u)) - B^{1}_{r} . [u_{mm} . -\wedge u] + J3^{T} [u_{max} - u] - \pi^{T}_{min}[x_{min} - x(u)] + \pi^{T}_{max}[x_{max} - x(u)].$$

C is initialized to the identity matrix and updated each iteration using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) updating formula, as suggested by Powell.

We also follow Powell's suggestion in calculating the step-size parameter, α:

Define $\Psi(\alpha)$ as:

$$\Psi(\alpha) = \Phi(u^{\#},x^{\#}) + \sum \mu_i \,|g_i(u^{\#},x^{\#})|$$

$$+ \sum \zeta_{i,max}\,|u_{i,max} - u_i| + \sum \zeta_{i,min}\,|u_i - u_{i,min}|$$

$$+ \sum \nu_{i,max}\,|x_{i,max} - x_i| + \sum \nu_{i,min}\,|x_i - x_{i,min}|$$

where

$$u^{\#} = u_k + \alpha d_k,$$

$$x^{\#} = x - \alpha(\partial g/\partial x^T)^{-1}\{g(u_k,x_k) + (\partial g/\partial u^T)_k d_k\}$$

and μ_i, $\zeta_{i,max}$, $\zeta_{i,min}$, $\nu_{i,max}$ and $\nu_{i,min}$ are defined as

$$\mu_i = \max\{|\lambda_i|\,,\ \tfrac{1}{2}(\mu^{*}_i + |\lambda_i|)\}$$

$$\zeta_{i,max} = \max\{|\beta_{i,max}|\,,\ \tfrac{1}{2}(\zeta^{*}_{i,max} + |\beta_{i,max}|)\}$$

$$\zeta_{i,min} = \max\{|\beta_{i,min}|\,,\ \tfrac{1}{2}(\zeta^{*}_{i,min} + |\beta_{i,min}|)\}$$

$$\nu_{i,max} = \max\{|\pi_{i,max}|\,,\ \tfrac{1}{2}(\nu^{*}_{i,max} + |\pi_{i,max}|)\}$$

$$\nu_{i,min} = \max\{|\pi_{i,min}|\,,\ \tfrac{1}{2}(\nu^{*}_{i,min} + |\pi_{i,min}|)\}$$

Note that values with a * indicate the value of that parameter at the previous iteration. λ_i is the current value of the Lagrangian multiplier for the equality constraints of the original problem, (P1). λ is calculated by:

$$\lambda = (\partial g/\partial x^T)^{-1}\{\pi_{max} - \pi_{min} - (\partial\Phi/\partial x)\}.$$

π_{max} and π_{min} are the Kuhn-Tucker multipliers associated with the inequality constraints derived from upper and lower bounds on the pivoted variables x and are calculated by the QPP associated with problem (P2). $\beta_{i,max}$ and $\beta_{i,min}$ are Kuhn-Tucker multipliers associated with the upper and lower bounds of the decision variables, and are also calculated by the QPP. At each iteration the value of *a* used is the first one found which satisfies VfclOftO). Usually *a* is 1, except perhaps for the first iterations.

4 THE ALGORITHM

Step 0: Initialization

i) Set k=0, 0^=1, /i$_Q$=0 (with C^I, the first direction predicted is the steepest descent direction).

ii) Initialize all variables z^Cx^u$_1$]

Step 1: Compute the Jacobian and Reduced Gradient

i) Increment k

ii) Evaluate *(d*/dz)*, g(z) and Og/3z^T) at z_R

iii) Perform forward Gaussian elimination to partition the Jacobian and find the L/U factors for (8g/dx^T)$_k$.

iv) Perform backward substitution to solve

Og/8x^T)[v_k, A_k] = [g, Og/3u^T)]$_k$

for [v_k, A_k]

and solve Og^T/3x)X*$_k$ = O*/8x)$_k$ for X*$_k$

v) Compute the reduced gradient

$$(\delta\Phi/\delta u)_k = (\partial\Phi/\partial u)_k - A^T_k(\partial\Phi/\partial x)_k$$

vi) If k<2, go to Step 3; otherwise go to Step 2.

Step 2: Update C (a la Powell's suggestions)

i) $\gamma_k = (\delta\Phi/\delta u)_k - A_k[\pi_{min} - \pi_{max}]_{k-1} - \omega_{k-1}$

ii) let $'_k = \theta\gamma_k + (1-\theta)C_{k-1}\delta_{k-1}$

where

$6 = 1.$ if $«^T_{k-1}{}'\gamma_k$ £ $.2a^T_{k-1}\ C_{k-1}{}^a i\wedge_{k-1}$

and otherwise

set

$9\ -\ {}^c{}_-{}^{8}{}_{1}{}^{T} f\ c\ -\ ,\ {}^c{}_{-1}] / [\delta_k\ V_1 C_{k-1}\delta_{k-1} - \delta^T_{k-1}\gamma_k]$

ill) Set $C_k = v_i - [C_{k-1}\delta_{k-1}\delta^T_{k-1}C_{k-1}]/[\delta^T_{k-1}C_{k-1}\delta_{k-1}]$

$+\ \wedge\gamma^T_k]/[\delta^T_{k-1}\gamma_k]$

Step 3: Solve Associated QPP

i) Define $Q(Au_k) = Q(x_k, u_k) + (a^*/«u^T)Au + \%Au^T_k C_k Au_k$

ii) Let d_k be the solution to the following QPP:

$$\min_{Au_k} Q(Au_k)$$

s. t. $- A_k Au_k$ £ $X_{MX} - x_k \bullet v_k$

$A_k \wedge Au_k \wedge S\ x_k - x_{min} - v_k$

$U_{min} A - U_k, \wedge Au_k, \wedge U_{max} - U_k$

Step 4: Compute Step Size Parameter, a_k

i) Calculate $X_k = Og^T/3x)"^1[*_{max} - v_{A\,min}] - X^*$

ii) Set $p_k = A_k d_k + v_k$

iii) Set $,_{ik} = \max\{|X_{ik}|, hi_{i(k-,)} * \backslash\backslash J)\}$

$\zeta_{ik,max} = \max\{|\beta_{ik,max}|, ½(\zeta_{i(k-1),max} + |\beta_{ik,max}|)\}$

$\zeta_{ik,min} = \max\{|\beta_{ik,min}|, ½(\zeta_{i(k-1),min} + |\beta_{ik,min}|)\}$

"ik.max = ""^'ik.n-xl' *{"1(k-1).max + Kk.maxl)}

$v_{ik,min} = \max\{|\pi_{ik,min}|, ½(v_{i(k-1),min} + |\pi_{ik,min}|)\}$

iv) Select $«_k G[O,l]$ such that $¥(x^*,u^*,//_k) < \Psi(x_k, u_k, \mu_k)$

where $¥(x,u,/i) = 4>(x,u) + //^T|g(x,u)|$

$$+ C_{max}^{T} | u_{max} - u_1 | + C_{win}^{T} | u_1 - u_{win} |$$

$$+ v_{max}^{T} | x_{max} - x_1 | + v_{m1n}^{T} | x_1 - x_{mm} |$$

$$x^* = \ \ " \ tfP_K$$

$$« \bullet - u_k + \alpha d_k$$

v) Set $\bar{x}_k = x^*; \ u_k = u^*; \ \delta_k = \alpha_k d_k$

vi) Let $i0_k = (\delta\Phi/\delta u)_k - A_k[\pi_{min} - \pi_{max}]_k$

Step 5: Clieclc for Convergence

If $(d^T_k <*_k + 9^T_k 9_k) < «$ then go to step 6;

otherwise go to Step 1

Step 6: Stop

The only differences between this algorithm and the one published by Berna et al. (1978 occur in Step 2 part i (calculation of y_k> and in Step 4 part i (calculation of

V-

5 DISCUSSION

It is of interest to compare this algorithm with both the Berna et al. 1980 algorithm and the one used in MINOS/AUGMENTED. We can do this by examining the final C* matrix that is approximated by the three algorithms, where C* is the Hessian matrix projected into the subspace of the decision variables u. The C* matrix for all three algorithms is given by:

$$C^* = [I \ -A^T] \begin{bmatrix} fo^2L/au3u^T) & @^2L/8u8x^T) \\ Ud^2Udxdu^J) & (d^2Udxdx^T) \end{bmatrix} \begin{bmatrix} I \\ -A \end{bmatrix}$$

where matrix A is calculated from

$$A = \{dgldx^J)\sim Hdgldu^T)$$

and

$$L(x,u,\lambda) = \Phi(x,u) - \lambda^T g(x,u)$$

and all functions are evaluated at the solution, (x*, u*, λ*). For clarification we are ignoring the terms added to the Lagrange function by upper and lower bounds. These terms drop out when second derivatives are taken.

In the Berna et al. 1980 version, the current point (x_1, u_1, λ_1) and the previous point (x_0, u_0) are used to update the matrix H_0 in such a way that H_1 satisfies the secant relationship, where H is an approximation to the Hessian matrix:

$$H_1 \begin{bmatrix} u_1 - u_0 \\ x_1 - x_0 \end{bmatrix} = \begin{bmatrix} \partial/\partial u[L(x_1, u_1, \lambda_1) - L(x_0, u_0, \lambda_1)] \\ \partial/\partial x[L(x_1, u_1, \lambda_1) - L(x_0, u_0, \lambda_1)] \end{bmatrix}$$

One of the variable metric method (VMM) update formulae is used to calculate matrix H_1. The desired H* is then

$$H^* = \begin{bmatrix} (\partial^2 L/\partial u \partial u^T) & \partial^2 L/\partial u \partial x^T) \\ (\partial^2 L/\partial x \partial u^T) & \partial^2 L/\partial x \partial x^T) \end{bmatrix}$$

evaluated at (x*, u*, λ*).

C_1 is constructed by computing

$$C_1 = [I \quad -A^T] \quad H_1 \begin{bmatrix} I \\ -A \end{bmatrix}$$

In MINOS/AUGMENTED, the current point (x_1, u_1, λ_1) is used to generate a new function, $f\{u \mid (x_1, u_1, \lambda_1)\}$:

$$f\{u \mid (x_1, u_1, \lambda_1)\} = \Phi\{(x_1 - (A)_1(u - u_1)), u\} - \lambda^T_1 g\{(x_1 - (A)_1(u - u_1)), u\}$$

where $(A)_1$ indicates matrix A evaluated at the point (x_1, u_1).

By using gradients of f at various values of u, the Hessian of f is approximated by VMM updating formulae. The Hessian of f at x_1 is given by:

$$\partial/\partial u)f\{u \mid (x_1, u_1, \lambda_1)\} = \partial/\partial u\{\Phi(x_1,u_1) - \lambda^T_1 g(x_1,u_1)\} + \partial/\partial x\{\Phi(x_1, u_1)$$
$$+ \lambda^T_1 g(x_1,u_1)\}(A)_1$$

$$0^2/au3u^T)f\{u_1| \ (x_{1\#} \ u_r \ X,)\} = [I, -(A),] \begin{bmatrix} \partial^2 L_1/\partial u \partial u^T & \partial^2 L_1/\partial u \partial x^T \\ \partial^2 L_1/\partial x \partial u^T & \partial^2 L_1/\partial x \partial x^T \end{bmatrix} \begin{bmatrix} I \\ -(A)_1 \end{bmatrix}$$

where L_1 is L evaluated at the point $(x_y \ u_y$ X^.

It is easily seen that for both of these algorithms the matrix A plays an important part in the approximation of C*. Under the assumption that f(u| $ix_y \ u_y$ X,) and L(x, u, X) are quadratic functions (actually that the quadratic approximation is locally sufficient), then the C matrix is a quadratic approximation to L(x*, u*, X*) projected onto the subspace where $x - x_1 = -(A\wedge u - u_o)$.

Now consider the algorithm given here. Examine the parametric programming problem:

$$P(u): \min_x \Phi(x,u)$$

$$g(x,u) = 0$$

Since it is assumed that g<x,u) has a unique solution x(u) for each u, P(u) has a unique optimal solution. Define $*(u) as the value of P(u). It is known that:

1. 34>*(u)/3u = $\partial\Phi(x,u)/\partial u$ - X^T(u)3g<x,u)/au

2. $\partial^2\Phi(u)/\partial u \partial u^T = t\backslash. \ -(A)^T] \begin{bmatrix} a^2L/8u3u^T & 3^2L/3u3x^T \\ a^2L/8x3u^T & 3^2U3x3x^T \end{bmatrix} \begin{bmatrix} I \\ -A \end{bmatrix}$

where x = x(u)

X(u) = -8/8x{g(x(u),u)}8/8x{4»{x(u)$_\#$u)}

L evaluated at the point (u, x(u), X{u)).

Note that C* = a^2{<f>*(u*)}/8u3u^T. If it is assumed that **(u) is quadratic, then a VMM update formula using $\partial\Phi^*(u)/\partial u$ at various values of u should construct an approximation to C^*. The algorithm presented here proceeds in this manner (except of course that x(u) may not be computed exactly). It attempts to construct C^* directly -- not as a quadratic approximation to the Lagrangian projected onto a subspace. Hence the curvature of x(u) is taken into account.

6 SAMPLE PROBLEMS

In this section we present the results of two test problems solved using the algorithm described in the previous section. The algorithm has been imbedded into the ASCEND-II flowsheet system (Locke, 1981). Modules describing these test problems were written and added to the system.

6.1 DETAILED SOLUTION OF A SMALL PROBLEM

We wish to solve the problem:

$$\min\{\Phi\}$$

$$\text{s.t.} \quad ab + bc - 1 = 0 = g_1$$

$$\Phi - a^2 - b^2 + c = 0 = g_2$$

$$0 \le a,b,c \le 1$$

Throughout the calculations variables Φ and a were the pivoted variables, while b and c were the independent (decision) variables. Initial values were: a=0, b=1, c=1, Φ=0. Also, at the start we set μ_0=0, and C_1=I.

The starting point gives an initial Jacobian Matrix of:

$$\begin{array}{c} \\ g_1 \\ g_2 \end{array} \quad \begin{array}{cccc} a & b & c & \Phi \end{array}$$
$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & -2 & 1 & 1 \end{bmatrix}$$

With Φ and a as pivoted variables, the matrix $(\partial g/\partial x^T)$ is the identity matrix.

The initial right-hand-side vector (g) is [0,0], so the vector v is also [0,0], while the matrix A is calculated to be:

$$\begin{bmatrix} 1 & 1 \\ -2 & 1 \end{bmatrix}$$

The initial QPP is:

min [2 -1] fAbi • Vz [Ab Ac] ["i Jiur
[ACJ »> 11K]

s.t. -1 £ Ab £ 0
-1 £ Ac £ 0

$$\begin{bmatrix} -1 \\ -\infty \end{bmatrix} \le \begin{bmatrix} 1 & 1 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} \Delta b \\ \Delta c \end{bmatrix} \le \begin{bmatrix} 0 \\ \infty \end{bmatrix}$$

The solution to this QPP is [-1,0]. The Lagrange multipliers on the original equality constraints are [0,-1]. Table 1 summarizes the search for *a* such that YU) < Y(0).

Table 1: Search for *a*

a	Φ	Za	Ψ
0.0	0.0	0.0	0.0
0.25	-0.375	0.125	-0.25
0.5	-0.5	0.5	0.0
0.75	-0.375	1.125	0.75
1.0	0.0	2.0	2.0

Using a step size of .25, the variable values after the first iteration are: a=.25, b=.75, c=1., 4>=-.5

Throughout the remaining iterations, c=1 and <x=1. The constrained derivative for iteration 2 is: £4>/£u=[-.6667, -1.5] and the Hessian Matrix after the first update is:

$$\begin{bmatrix} 5.33 & 2. \\ 2.0 & 1.891 \end{bmatrix}$$

Table 2 shows the progress of the calculations to the solution.

Table 2: Progress of the Calculations

Iteration	a	b	RHS	•
0	0.0	1.0	0.0	0.0
1	0.25	0.75	0.0988	-0.5
2	0.5417	0.6250	0.0757	-0.4170
3	0.3910	0.7101	0.0232	-0.3738
4	0.3947	0.7170	5.4E-5	-0.3301
5	0.3308	0.7498	3.9E-3	-0.3345
6	0.3780	0.7248	2.2E-3	-0.3346
7	0.3815	0.7238	9.6E-6	-0.330514
8	0.3807	0.7242	5.7E-7	-0.330501
9	0.3803	0.7245	2.1E-7	-0.330501
10	0.380278	0.724492	7.6E-13	-0.330500

6.2 OTHER COMPUTATIONAL EXPERIENCE

The algorithm has been applied to several flowsheeting problems. The largest contained 156 generally non-linear equations with 161 calculated variables, leaving 5 degrees of freedom. The algorithm was able to find the optimal solution to this problem without difficulty. Typical convergence was in 10 to 20 iterations for all problems.

7 ACKNOWLEDGMENT

This work was partially funded by NSF Grant ECS-8026777 (Edahl) and by a grant from the Exxon Corporation (Locke and Westerberg).

REFERENCES

[I] Berna, T.J., M.H. Locke, and A.W. Westerberg.
Optimization of Large Chemical Process Systems.
In Savage, G.J. and P.H. Roe (editors). *Large Engineering Systems* 2. Sandford Educational Press, May, 1978.

[2] Berna, T.J., M.H. Locke, and A.W. Westerberg.
A New Approach to Optimization of Chemical Processes.
AIChE Journal 26(1h37-44, January, 1980.

[3] Biegler, LT. and R.R. Hughes.
Approximation Programming of Chemical Processes with Q/LAP.
Chemical Engineering Progress **, April, 1981.**

[4] Biegler, LT. and R.R. Hughes.
Infeasible Path Optimization with Sequential Modular Simulators.
AIChE Journal **, 1982.**
Accepted for Publication.

[5] Box, MJ.
A New Method of Constrained Optimization and a Comparison with Other Methods.
Computer J. 8(1):42-52, 1965.

[6] Edahl, R.H.
Sensitivity Analysis in Nonlinear Programming: Applications to Comparative Statics and Algorithmic Construction.
PhD thesis, Carnegie-Mellon University, May, 1982.

[7] Friedman, P. and K. Pinder.
Optimization of a Simulation Model of a Chemical Plant.
Ind. Eng. Chem. Proc. Des. Dev. **11:512, 1972.**

[8] Han, S.P.
A Globally Convergent Method for Nonlinear Programming.
Technical Report 75-257, Cornell University, 1975.

[9] Locke, M.H.
A CAD Tool which Accommodates an Evolutionary Strategy in Engineering Design Calculations.
PhD thesis, Carnegie-Mellon University, September, 1981.

[10] Murtagh, B.A. and M.A. Saunders.
MINOS/AUGMENTED User's Manual.
Technical Report SOL 80-14, Stanford University, 1980.
Systems Optimization Laboratory.

[II] Murtagh, B.A.
On the Simultaneous Solution and Optimization of Large-Scale Engineering Systems.
Computers and Chemical Engineering **6(1): 1-5, 1982.**

[12] Powell MJ.D.
A Fast Algorithm for Nonlinearly Constrained Optimization Calculations.
Presented at 1977 Dundee Conference on Numerical Analysis.
1977.

[13] Robinson, S.M.
Perturbed Kuhn-Tucker Points and Rates of Convergence for a Class of Nonlinear Programming Algorithms.
Math Prog. :1-16, 1974.