Department of Chemical Engineering Carnegie Institute of Technology

1989

Reduced SQP implementation for large-scale optimization problems

Sriram Vasantharajan
Carnegie Mellon University

J Viswanathan

Lorenz T. Biegler

Carnegie Mellon University.Engineering Design Research Center.

Reduced SQP Implementation for Large-Scale Optimization Problems

by

S. Vasantharajan, J. Viswanathan, L.T. Biegler

EDRC 06-66-89
Carnegie Mellon University

REDUCED SQP IMPLEMENTATION FOR LARGE-SCALE OPTIMIZATION PROBLEMS

Sriram Vasantharajan[*], J. **Viswanathan**[+] **and** L.T. **Biegler**[*,+]

[*] *Chemical Engineering Department*
[+] *Engineering Design Research Center*
Carnegie Mellon University
Pittsburgh, PA 15213

Financial support from the Engineering Design Research Center, an NSF sponsored center, and the Department of Energy is gratefully acknowledged for this study.

Abstract

The development and implementation of the Range and Null space Decomposition (RND) strategy for large-scale problems is described with emphasis on the optimization of engineering systems. The RND technique, as detailed in Vasantharajan and Biegler (1988), uses nonorthonormal, gradient based projections for the Jacobian. However, this implementation is dense, and does not take advantage of system sparsity. Here we extend this algorithm to incorporate general purpose sparse matrix techniques. Also, problems like inconsistent linearizations and infeasible Quadratic Programs (QPs), which are generally associated with QP based methods compromise the robustness of this method and need to be considered. Finally, systematic ways of generating a nonsingular basis for general nonlinear programs must be developed if this strategy is to be adapted to solve large, sparse problems efficiently. To deal with these problems, a two phase LP-based procedure is coupled to the RND algorithm. This strategy also serves to partition the variables into decisions and dependents, thereby generating a nonsingular basis. Any redundancies/degeneracies in the constraints are also detected and processed separately. The entire reduced SQP implementation is then interfaced with GAMS (Brooke *et a/.* (1988)), a front end for representing and solving process models.

Finally, a thorough comparison of the RND based reduced SQP strategy with MINOS (Murtagh and Saunders (1978)) is effected on a set of NLPs and process design problems. The process problems include the optimization of the operation of distillation columns. These problems warrant special mention as have been uniquely conceived and implemented in a novel equation-oriented manner, thus exploiting the full potential of the GAMS architecture. Detailed discussion of the formulation and results are included and results are obtained that confirm the viability and efficacy of the reduced SQP implementation for efficient solution of large, difficult nonlinear programs.

1. Introduction

Design systems are in general described by large models, requiring the solution of large, sparse sets of equations depicting the process modules and the engineering operations, but only a few degrees of freedom. For example, for a typical flowsheet the number of variables and equations may be of the order of 10,000, while the parameters would be of the order of 10 (Locke *et a/.* (1983)).

Although there is still no universal consensus about the "best" algorithm for nonlinearly constrained optimization in the dense case, SQP has been recognized by many studies (Reklaitis et al, 1983; Hock and Schittkowski, 1980; Edgar and Himmelblau, 1988) as one of the most efficient algorithms for small to moderately sized problems, requiring the fewest function evaluations for optimization. However, as the size of the problem become large, it has been observed to be time consuming. The computational effort and overhead required to solve QPs with current, dense quadratic programming implementations become significant and, indeed, may be the major computational cost.

Unlike SQP, MINOS was developed as an efficient nonlinear programming algorithm for large-scale implementations. In this approach, the nonlinearly constrained problem is solved as a sequence (major iterations) of linearly constrained sub-problems, which have as their objective function a projected Lagrangian. The solution of each linearly constrained problem can be found by applying unconstrained algorithms (minor iterations), such as variable metric methods, in the null space of the active linear constraints. Under mild conditions it can be shown (Murtagh and Saunders (1978)) that the major iterations converge to the solution of the nonlinear program at a quadratic rate. Minor iterations generally converge to the solution of the linearly constrained problem at a superlinear rate. Moreover,-although this strategy generally requires more function and gradient evaluations than SQP, it lends itself to efficient sparse implementations, especially with respect to matrix factorizations for null space operations. Consequently, the MINOS algorithm often performs well on large scale process optimization problems.

With the Range and Null space projections, on the other hand, a reduced SQP algorithm, RND, has been developed, giving SQP the potential to solve large, sparse problems. RND solves a much smaller quadratic program only in the null space of the active constraints, which is of the dimension of the degrees of freedom of the problem. Linear equations then need to be solved to compute the range space move to a feasible point on the linear subspace of the active constraints. In addition, this algorithm has been shown (Nocedal and Overton, 1985) to have the same theoretical convergence properties as the that of the undecomposed SQP algorithm. Moreover, our computational experience (Vasantharajan and Biegler, 1988) has shown that this method almost never requires more iterations than the full SQP, and only a small fraction of the computational cost.

Even for large scale problems, however, effort required for function evaluations is still significant, and therefore, SQP strategies tailored to these problems still can be very effective. Thus, there is considerable incentive to develop a reduced SQP strategy through range and null space decomposition for process optimization problems. It should be noted here that the measure of efficiency of any algorithm for solving large-scale process optimization problems should be based both on the number of function or model evaluations, as well the computational time required.

2. Adapting RND for Process Optimization Problems

With the fact that process optimization problems usually have many variables and equations, and relatively few degrees of freedom in mind, let us re-examine the final system of reduced space equations as proposed by the RND approach:

Null space Direction

$$Min \quad [V«\underset{u}{t}>^T - V_v 4>^T ft] \; d_z + \frac{x}{} d^*_z [Z^T B \; Z] \; d_z \qquad \text{(RND)}$$

$$st. \quad [V / - V / ft] \; d_z \wedge - g - [V / ft^T + V g^T] \; d_Y$$

Range Space Direction

$$\text{with} \quad d_y = - (I + \alpha\alpha^T)^{-1} \beta,$$

$$x = - (V_v h)'^1 \; (I + oca^1)'' \; Y^T \; (V^* + Vg \; \mu)$$

$$d = \begin{bmatrix} d_z \\ -ad_z \end{bmatrix} + \begin{bmatrix} a^T d_Y \\ d_Y \end{bmatrix} \qquad \text{(D}$$

Here, u and v denote the decision and dependent variable subsets of the set process variables, x; d denotes the search direction in x and *\L* are the multipliers of the inequality constraints from the reduced space QP (RND). X are the multipliers that correspond to the equality constraints. Also, a has been defined as the matrix $\lfloor f(V_v h^1)''^1 \; V_u h^T \; 1 \rfloor$ and j3 as the vector $\lfloor f(V_v h^1)''^1 \; hi \rfloor$ for convenience of notation. From these we can define the range space projection matrix as $Y = [\; a \quad I \;]^T$ and the null space projection matrix as $Z = [\; I \quad -ft^T \;]^T$. The matrix $(I + tttt^T)$ can be seen from the definition of a to be of the same size as the number of equality constraints, h, in the problem. This makes the LU decomposition task for this matrix computationally prohibitive. However, this matrix can be reduced using *Householder's* transformation as follows:

$$(I + aaV = [i - a (I * a^T a \; r^1 \; a^T] \qquad \text{(2)}$$

Now, the matrix to be decomposed, $(I + a^T a)$ is only of the size of the number of degrees of freedom in the problem, and hence denotes a very significant reduction in terms of computation required. With this transformation, the modified expressions for d_Y and X can be written as follows:

$$d_Y = -\left[\beta - \alpha\ (I + \alpha^T\alpha)^{-1}\ (\alpha^T\beta)\right].$$

and,

$$\lambda\ = -\ (\nabla_v h)^{-1}\left[\ I - \alpha\ (I + \alpha^T\alpha)^{-1}\ \alpha^T\ \right] Y^T\ (\nabla\Phi + \nabla g\ \mu) \qquad (3)$$

The reduced space algorithm then consists of solving for the range and null space directions at each iteration and then performing a linesearch along the search direction, d. A detailed description is given in a later section. To deal with large, sparse Jacobian matrices, sparse routines from the Harwell subroutine library, viz. MA28AD and its related procedures, are used to solve the systems of linear equations to construct the matrix α. The routine MC19 is employed to obtain the scaling factors for the matrix to be factored. On the whole, the Harwell library routines seemed adequate for this implementation.

3. Consistent Linearization and Basis Generation

The SQP algorithm solves a sequence of QPs, the constraints of which are obtained by a 1st order Taylor's series expansion of the nonlinear constraints imposed on the problem. Denoting the equality constraints by "h", and the inequality constraints by "g", this can be represented as follows:

$$h(x) = h(x_k) + \nabla h^T(x_k)\ (x - x_k) = 0$$

$$g(x) = g(x_k) + \nabla g^T(x_k)\ (x - x_k) \le 0 \qquad (4)$$

A frequent problem in solving quadratic programming sub-problems is that the quadratic programming problem does not have a solution. This occurs when the linearized system of equations, as shown in eqn. (4) is inconsistent. A necessary and sufficient condition for this is that the rank of the coefficient matrix be equal to the rank of the matrix augmented with the right-hand side. For large-scale problems where "good" starting points are not available, the probability of encountering infeasible

QP iterations is high. In the past, safeguards for SQP (Biegler and Cuthrell (1985)) have been proposed, that have the effect of relaxing the feasible region, and thus solving an altered QP. However, none of these relaxation techniques avoid failure on all problems.

Here, we will propose a systematic approach to handling infeasible quadratic programs, which parallels the *phase* I strategy adopted by many LP strategies to find an initial feasible basic solution. It is formulated as an LP, with each equality constraint padded with two non-negative artificial variables, and each inequality constraint augmented by one such variable. Denote these additional variables as P and W for the equalities, and by S for the inequalities. The objective of this auxiliary linear program is to minimize the sum of these artificial variables. The expanded system, which we will call *primary phase* I LP, can be written as:

$$\mathit{Min} \quad \overset{\mathbf{MEQ}}{T''} \; < P_i + W_i > \; + \; \overset{\mathbf{IEQ}}{X} \; S_j \qquad\qquad <LP1>$$

$$\text{st.} \quad h_i(x_k) + Vh_i^T(x_k)\;(x - x_k) = P_i - N_i$$

$$g_j(x_k) + Vg_j^T(x_k) <x - x_k) \le S_j$$

$$x^L \pounds (x - x_k) \le x^u$$

$$P_i, N_i, S_j \pounds 0 \quad V \; i.\; j$$

Here, MEQ and IEQ denote the number of equality and inequality constraints, respectively. At the LP solution, the value of each of these artificial variables measures the magnitude of the infeasibility in its associated constraint If there is a feasible solution to the original system of constraints given by eqn. (4), then it is obvious that (LP1) has minimum value of zero, with P, N, and S, each equal to zero. Thus, if (LP1) terminates with all these auxiliary variables non-basic at zero, then the the set of equations is consistent. If however, any artificial variable is basic at a non-zero value, implying infeasibilities in the linear system, the LP solution is then used as a search direction for the next point. A step length, CO, is then computed along this direction for which some merit function, measuring the constraint infeasibility is minimized. We have chosen the Ll-norm of the constraint values

$$\theta(x) = \sum_{i \in I} |h_i| + \sum_{j \in J} \max[0, g_j] \qquad (5)$$

as this merit function. Here, the set I is the set of all MEQ equality constraints, and J the set of IEQ inequality constraints. Partitioning the set of equality and inequality constraint each into three set as follows:

$$I_1 = \{i \mid h_i > 0\} \qquad J_1 = \{j \mid g_j > 0\}$$
$$I_2 = \{i \mid h_i = 0\} \qquad J_2 = \{j \mid g_j = 0\}$$
$$I_3 = \{i \mid h_i < 0\} \qquad J_3 = \{j \mid g_j < 0\}$$

with

$$I = I_1 \cup I_2 \cup I_3,$$
$$J = J_1 \cup J_2 \cup J_3 \qquad (6)$$

it can be shown (Vasantharajan, 1989) that the search direction given by (LP1) is a descent direction for the merit function *6.* An Armijo line search along this direction using *6* has been incorporated with the *primary phase* I strategy to reduce the sum of infeasibilities. At the new point computed, the constraints are once again linearized and a new (LP1) solved. This process is continued until the sum of the infeasibilities is below a chosen tolerance. This procedure is re-executed each time an infeasible QP is encountered during the intermediate iterations of the RND algorithm.

A principal advantage of embedding a full set of artificial variables into the constraint system is that any redundancy or degeneracy associated with the system does not pose computational problems. For example, if a given system has been over-specified, whereupon, a particular equation can be constructed by a linear combination of the others, it leads to a linearly dependent set of constraints, and therefore a singular system. With the Newton's method the main drawback is encountering singular Jacobian matrices. The parallel in the quasi-Newton method, RND, is a singular basis.

As linear dependence among the linear constraints is commonplace in practical problems, it is desired to have a suitable formulation to pick a non-singular set of dependent variables, as well as identify any redundancy in the set of equality constraints. An alternate *phase* I LP, which will be referred to as the *secondary phase* I, is devised for this purpose as follows:

$$\mathit{Min} \quad \overset{\text{MEQ}}{\text{T}*} \; (P + N) \qquad \text{(LP2)}$$

$$\text{st.} \quad h_i(x_k) + Vh_i^T(x_k)\,(x - x_k) = P_1 - N_1$$

$$P_i, N_i \text{ £ } 0 \;\; V\, i$$

This problem is identical to (LP1) sans the inequality constraint and the bounds on the process variables.

As solution of (LP1) has already ensured consistency of the entire system of constraints, we know that (LP2) is always feasible, with objective function value lower than the chosen tolerance. Further, with full set of artificial variables the equations are alway linearly independent by construction. If (LP2) terminates with all these variables non-basic at their lower bound, there are no redundant constraints. But, if any P_i or N_i is basic at zero, i.e. a degenerate solution, it signifies that the associated constraint is a redundant (but consistent) equation. Tjie offending constraint is not used in the computation of the range space step, but is imposed as a constraint on the QP, when computing the null space movement. The reason is that the degeneracy could be a local property at the current point, and the constraint could be a valid one at optimum. In addition, as part of the (LP2) solution, the LP solver identifies a set of basic process variables, v, commensurate with the number of basic or non-redundant equations.

Caution should be exercised if the Jacobian becomes singular at later iterations, since the effect of a change in basis on the projected Hessian should be considered. One obvious option is to reset the projected Hessian to Identity and restart with the new basis by resolving (LP2). This was found by experience to be effective, serving only to slightly slow down the method. A second option is to repartition the variables, but use the old projected Hessian. Although this seems to work on some of the problems we tested, it is not recommended. The third and probably most correct option is to compensate for the change in basis, by considering explicitly the influence of the basis Z on the update. This requires an explicit form for the transformation matrix M which permits the nonorthonormal basis to be expressed as a linear combination of the columns of an orthonormal basis, *i.e.*,

$$Z = \bar{Z}\ M \tag{7}$$

This can be computed by:

$$Z\ (Z^TZ){\sim}^{W2} = Z\ (M)''^{1} = \bar{Z} \tag{8}$$

and would require the storage of the basis $\bar{Z}$ for each iteration and a *Cholesky* factorization to compute M. Although we did not implement this approach, we mention this option here for completeness.

It should be noted here that in a purely equality constrained or inequality constrained problem solving (LP1) will suffice, and (LP2) is not necessary. Also, in a mixed problem, if the optimal solution 'of (LP1) has all the logical variables (slacks) associated with the inequality constraints in the basis, this impies that these constraints are not active, and it is not necessary to solve (LP2). On the other hand, even if one of the inequality constraints is active, then one of the process variables will be basic with respect to this equation. As it is not possible in sparse LP solvers to obtain a one-to-one correspondence between the basic variables and the basic constraints, it is necessary to solve (LP2) in the absence of this constraint to sort out the non-singular basis to the set of equality constraints.

If at intermediate iterations infeasible QPs are encountered, we return to the solution of (LP1), followed by (LP2), if necessary. If a singular basis is met, we return to (LP2).

4. Interface with GAMS

General Algebraic Modelling System (GAMS) (Brooke *et* a/., (1988)) is a high-level language for compact representation of large and complex models. It was developed primarily to make easy the onerous task of model representation and modification, and its subsequent optimization by mathematical programming techniques. The GAMS model portrait permits mathematical representations, and all data transformations are specified succinctly and algebraically. Of principal importance to us was the fact that GAMS compiler computes the objective and constraint function derivatives of a given model using symbolic differentiation. It also permits existing algorithmic methods to be interfaced efficiently, enabling the use of the GAMS framework to test NLP solvers.

The large-scale reduced SQP method has been interfaced with the GAMS system as a general purpose nonlinear solver. Unlike MINOS which has been specially adapted and integrated within the GAMS, SQP had to be interfaced within the bounds provided by the system's framework. Even so, as will be borne out by the computational results, the development has been a successful one.

The RND/GAMS implementation is a modular one and uses different commercial routines for operations such as LP and QP solutions, and solving systems of equations. For example, the FORTRAN code QPSOL (Gill et al, 1983) is used to solve the QP subproblems: the Harwell routine MA28 is used to factor the sparse Jacobian. However, care has been taken to ensure that the RND strategy is not wed to any of these routines. A flexible platform has been provided, whereby the incorporation of new and more efficient procedures is straightforward, and does not require a overhaul of the software.

5. Reduced SQP Algorithm

The RND algorithm, as refined and extended to solve large-sparse problems can now be detailed as follows:

- Set iteration counter k to 0.

- Preprocessing Phase:

1. Evaluate the functions and gradients 4>, V*, h, Vh, g and Vg at x_k. These are computed by GAMS in a sparse form with only the nonzero entries being processed.

2. Set up the *primary phase* I problem (LP1) and solve the linear program. Currently, any LP routine interfaced with GAMS (such as BDMLP, ZOOM, MPSX or even MINOS) can be used for this step.

3. If the LP1 objective function value is not less than the desired tolerance (10"^{6} default) then perform an Armijo line search along the search direction, d_k provided by the LP. Use the merit function *8* Ceqn. (6)] to find a step size CO.

   Update $x_{k*l} = x_k + d_k$. Increment counter: k = k + 1. Go to 1.

4. If all constraints are either equalities or inequalities, then stop. If we have a mixed set of constraints, and if all slacks associated with inequality constraints are basic, then go to step 5. Else, set up *secondary phase* I linear program (LP2) with only the equality constraints and solve.

5. Identify the basic variables and constraints. Tag the redundant equality constraints for special prncessing.

- Reset iteration counter. k = 0.

- Initialize the projected Hessian approximation to Identity matrix, **I**.

- <u>**Optimization Phase**</u>:

1. Set and solve a sparse system of linear equations using MA28 and related routines. Compute α and β by matrix multiplication. If basis singular, return to **Preprocessing Phase** to execute the *secondary phase* I procedure.

2. Set up system of equations for solution as an augmented matrix, denoted by C:

$$C = \left[(I + \alpha^T\alpha) \;\vdots\; \alpha^T\beta \right]$$

From the system C, compute using LU decomposition and matrix multiplication d_Y [eqn. (3)]

3. Solve the QP (RND) with the redundant equality constraints included, to obtain d_Z and μ. If QP is infeasible return to Preprocessing Phase to re-execute (LP1)

4. Reconstruct the search direction d and estimate the Lagrange multipliers for the equality constraints, λ [eqn. (3)].

5. If the error in the first order necessary conditions, the fractional change in the variables and objective function are each less than the desired tolerance, then stop. Else:

6. Perform an Armijo line search along the search direction d to get the step length, ω.

7. Set $x_{k+1} = x_k + \omega$ d. Set k = k+1.

8. Update the projected Hessian (Z^TB Z) using the BFGS update formula with Powell's (1977) positive definite correction.

6. Computational Experience

The sparse reduced SQP/RND implementation has been extensively tested against MINOS on a number of nonlinear programs, within the framework of the GAMS system. This software evaluation endeavor attempted to answer the following questions:

1. How does the reduced SQP strategy generalize to large, sparse problems?

2. How well does the RND strategy compare to MINOS in terms of number of function evaluations and computational effort?

3. How do the two procedures compare from the point of robust performance on difficult process optimization problems?

The performance comparison of the RND implementation, written in FORTRAN 77 was carried out against GAMS/MINOS version (5.1) on a VAX 8810. The VAX/VMS interface of GAMS/MINOS version (5.2) was not supported by GAMS for this hardware at the time of testing, so we decided to use the earlier version, to provide a consistent environment for drawing conclusions. However, the version (5.2) which was available with another version of GAMS implemented on a SUN 3/60 was used, whenever the earlier version proved inadequate for our purpose.

While tabulating the results, the following notation will be adopted throughout: N will be used to denote the number of variables, M the total number of constraints, and MEQ the number of equality constraints. These will be used to characterize the problems specifications. TIME" will be used to represent the CPU time required. Unless otherwise specified, the time unit is CPU seconds on the VAX 8810. TUNC" will depict the number of function evaluations required. When the algorithm fails of a particular problem the notation (F) will be employed to indicate this. Finally the heading "LP" shows the CPU time for the RND Preprocessing Phase. Here BDMLP, a sparse LP solver, was used as it was the most reliable LP option available* at time of testing.

The nonlinear problems considered can be classified into two distinct groups: The first class of example problems, which will be referred to as "General test problems", are from a repository which is included with the GAMS system. These are models collected from the early stages of development of GAMS. The results and sizes for these problems are presented in Table 1. As can be seen from this table, we have considered a wide variety of problem sizes, ranging from small « 50 variables) to medium (about 100) to large (> 200), primarily to address the first question posed above. These results will serve to illustrate that the transition to solving larger problems with a reduced SQP implementation has been an efficacious one.

The second set of problems are optimization of separation-based processes. The classical phase-contacting operations, absorption and distillation, are considered here. These problems, unlike the first coterie, are complex process models, and the optimization problems are not easy. The results for these are given in Table 2, and will address the second and third factors considered above. Based on the trends in these results, it is patent that the range and null space decomposition strategy is indeed a highly efficient and robust strategy for large-scale problems.

6.1. General Test Problems

These nonlinear problems from the GAMS library are real world models. This library is an exhaustive one, and contains models from diverse are of applications like, Agricultural Economics, Engineering, Chemistry and Chemical Engineering, Mathematics etc. For further details of these models and their origins, refer the GAMS, A Users Guide (Brooke *et* a/., (1988)). A brief outline of each of these problems will be given here. A feature of all these problems is that they are tractable, and have known optimal solutions.

The first problem, "Ramsey", is a savings model which illustrates the trade-off between consumption and investment The second, "Chenery", is also an economic development model which deals with substitution and structural change. These two constitute the "small" example problems. The last three examples are from the field of applied general equilibrium. While "Korcge" and "Camcge" are models for Korea and Cameroon respectively, "Ganges", is a macroeconomic framework for India.

Both MINOS (5.1) and RND are successful in solving all these problems. RND, as expected, always requires fewer function evaluations than MINOS. On the two smaller problems, RND is better than MINOS in terms of the computational effort required. Although the function evaluations for the reduced SQP are considerably lower, on these simple models, the effect on CPU time is negligible. On the medium-sized problem, "Korcge", and on one of the larger problems "Camcge", RND requires less than half the CPU time taken by MINOS, although the differential in the number of function evaluations is not significant. The last problem in this category, "Ganges", is unusual from an engineering perspective in that it has 83 decisions out of a total of 357 variables, a very significant number. Consequently, although RND requires only 4 iterations, the solution of each of these large QPs is still time consuming. This translates into the reduced SQP strategy requiring about 20% additional CPU time than MINOS. However, with the inclusion of a more efficient QP routine, the solution of these large problems should be much faster.

These results confirm the distinct advantages that the RND approach offers to solve general nonlinear programs. On this set of varied models solved, RND is computationally very competitive with MINOS, and always requires fewer model evaluations.

PROBLEM	CO a IFICATION8			MINOCO (5.1)		LUOUOLUQ SQP		LP
	N	M	MEQ	TIME*	FUNC	TIM•UJ	FUNC	TIMUJ*
RAMSEY	CO4	CO CM	10	ay o	o ID	0.7	10	0.7
CHENERY	44	O) CO	o CM	1.7	CM	1.5	100	1.9
KORCGE	1oo	CO CD	78	4 CM	LO	1.0	8	p CVi
CAMCGE	o 00 CM	CO CM	CO CM	1LO.IT)	9	8.1	8	1in.
GANGffi	LO , CO	h ,4 CM	r˜/4 CM	1CD.CM	1CM	CVJ1.1	4	14.CM

* OoU CO 8 td to - VAX 8810

Table 1: Compuutional Results for General Nonlinear Problems

6.2. Process Optimization Problems

The second class of problems considered are separation process optimization problems involving absorber and distillation models. These models exhibit the conventional design problem characteristics of being described by a large number of variables and equations. In all four major models/systems are considered here, and the optimization results are presented in Table 2. The first one is an absorber model and the last three are distillation columns modelled from first principles. Of the distillation problems, the first one is a design specification problem, which assumes ideal vapor and liquid properties. A brief discussion of these two models will be presented. The last two distillation problems are optimization of the operating conditions. These are realistic and complex models, with rigorous thermodynamic equations for the vapor and liquid phase properties. The implementation details, and an in-depth discussion of the solutions obtained for these two models will be presented here. As these large models are computationally intensive to evaluate, and the overlaid optimization problems are difficult to solve, they will serve as apposite tests to evaluate the efficiency and robustness of the optimization algorithms compared.

The first example is the optimization of an absorber process model described by Kremser equation (refer to Kocis (1988) for details). This process unit model neglects heat effects and assumes a pure solvent The objective here is to minimize the sum of the capital cost, expressed as a function of the number of trays in the absorber, and operating cost, calculated as a function of the stream and component flowrates. Two cases have been considered in this model; these are denoted as Absorber (a) and (b) in Table 2. In the first case, the absorption factor is a decision variable, and an upper bound of 20 was imposed on the number of trays. In the second case, the model was simplified by fixing the absorption factor and increasing the maximum number of plates to 30. As can be seen from the results, on the first problem MINOS (5.1) terminated at an infeasible solution, although the (5.2) version was successful in finding the optimal

PROBLEM	SPECIFICATIONS			MINOS (5.1)/(5.2)		REDUCED SOP		LP
	N	M	MEQ	TIME*	FUNC	TIME*	FUNC	TIME*
Absorber	50	42	42					
(^{a})				(F)/26.3$^{\#}$	(F)/140	1.93	33	0.82
(b)				23	91	1.11	14	0.82
Distillation Ideal	228	227	227					
(a)				17.8	12	11.9	7	73.5
(b)				26.1	85	19.9	12	85.0
Distillation Nonideal (1)	569	567	567					
(a)				115.9	43	35.2	14	124.3
(b)				231.9	289	94.1	48	12.3
(c)				(F)/1.6^{+}	(F)/746	96.7	55	105.6
Distillation Nonideal (2)	977	975	975					
(a)				(F)/(F)	(F)/(F)	104.1	31	365.5
(b)				360.2	162	190.8	26	27.7
(c)				(F)/(F)	(F)/(F)	193.4	34	316.2

* CPU Second-VAX 8810

+ CPU Hours SUN-3/60

(F) Failed

# CPU Seconds SUN-3/60

Table 2: Computational Results for Process Optimization Problems

solution. On the second problem, version (5.1) proved adequate. RND was able to find the optimal solution in both cases with less computational work. In addition, in keeping with the forte of this approach, very few model evaluations are needed, a small portion of the calls required by MINOS.

Problem specifications for the next three problems are given in Vasantharajan (1989). The first of these is denoted as *Distillation Ideal*, and the last two cases as *Distillation Nonideal (1)* and *Distillation Nonideal (2)*, respectively. In the results given in Table 2, individual cases considered for these will be indexed by (a), (b) etc. The convention adopted here is that for the nonideal problems, case (a) would denote the simulation result, case (b) the subsequent optimization, and case (c) the results for a simultaneous solution technique.

All these columns were initialized with linearly interpolated starting points, using the simulation specifications given in the literature. A linear pressure distribution, with a pressure drop of about 0.1 atm was assumed along the column. The symbols, conventions and equations used in describing the columns, as well as the mathematical model of these columns can be found in Vasantharajan (1989).

Problem *Distillation Ideal* deals with the separation of a binary mixture of benzene and toluene and is required to determine the minimum external reflex ratio to effect a specified separation for a given number of stages. Both the liquid and vapor phases were assumed to exhibit ideal behavior, a reasonable assumption for this system at the column pressure. Two different starting values for the reflux ratio were tried out. The second case was purposely initialized far away from the optimal value predicted from the first starting point. This is reflected in the performance of both the optimization algorithms, which require more effort for the second case. The previous trends observed in comparing MINOS and RND are also seen here. In both cases, the reduced SQP implementation outperforms MINOS.

6.2.1. A New Equation-Oriented Model for Distillation

The motivation to solve the last two nonideal separation models, interestingly, stemmed from the research efforts of Lucia and coworkers. As an alternative to the use of decomposition strategies to improve the performance of SQP, Kumar and Lucia (1987) developed sparse Hessian updates, tailored to distillation optimization problems. These update formulae exploit the properties of thermodynamic functions and the sparsity in the Hessian matrix. In this hybrid approach, derivative information is supplied where calculable, and remaining information is approximated through sparse quasi-Newton formulae that are thermodynamically consistent (TCH). It has been observed by Lucia and Kumar (1988) that in evaluating the numerical performance of quasi-Newton methods on distillation optimization example problems, that feasible starting points were mandatory. All the methods employed failed when a simultaneous simulation and optimization approach was adopted to solve these models, using linearly interpolated starting points. Rather, they claimed that a sequential approach was necessary, which simulates the column first, and then optimizes the operating conditions. Further, even when feasible starting points are used, only the TCH method was able to successfully solve all the problems considered. The authors then stated that it would be interesting to examine the performance of RND on these arduous problems.

It should be noted here that although significant improvements in performance have been noted with TCH, no theoretical improvements in convergence have yet been shown, and, in fact, computational difficulties can occur because the updated Hessian may not be positive definite. Moreover, these strategies apply only to specific classes of problems with special structure. For general purpose problems, it may be difficult, if not impossible, to develop sparse updating formulae that improve the performance of SQP. However, accepting the challenge, we decided to model and solve two of the example problems used by Lucia *et* a/.,.which, in particular, exemplified the reliability and efficiency of the TCH method.

In order to exploit the facilities in GAMS, we decided to adopt a novel approach to describe these detailed and rigorous models. In the conventional "sequential-modular'* approach, which is adopted by almost all commercial process simulators, instead of specifying the full process model as a single set of nonlinear equations that need to be solved, process equations are grouped according to the particular unit operation they describe. Physical and thermodynamic properties required are evaluated by calling the appropriate procedures. While the modular nature of the formulation enables easy construction and modification, gradient information is not directly available. As accurate computation of these gradients is imperative for optimization, we decided to embed the thermodynamic package in the distillation model, resulting in an equation-oriented representation of the entire problem. By doing so, the GAMS symbolic differentiation capability can then be used to compute all the necessary gradient information analytically. Although this approach would lead to a large increase in the number of process variables, it should be noted here that all the physical property related variables are fully determined by the equations defining them. Consequently, by using the reduced SQP technique, all these equations can be decomposed. Thus, the final optimization problem solved will be no larger than the one that would be obtained by decomposing a model described in a modular environment

The ease with which the amalgamation of the physical property equations and the model description was achieved is a tribute to the excellent facilities for data and model representation in GAMS. The liquid phase was modelled by UNIQUAC (Prausnitz *et al.* (1980)) method. The generalized method of Hayden-O'Connell was used to compute the pure component and cross second virial coefficients to evaluate the vapor phase fugacity coefficient The enthalpies of both vapor and liquid were corrected for the effects of pressure and mixing. The entire system representation was divided into five files/sections of information, in such a way that the development of the model would parallel the hierarchical manner in which an individual would conceptually decompose these problems. The first section was the physical property database, where, all the required properties, parameters and constants for the process components were tabulated and computed. This file was then compiled by GAMS and saved. In a second file, the vapor phase related parameters and their definitions were represented. This file was then processed and saved, restarting from the saved data from the first file. A third file, which had the liquid phase related particulars, was then similarly processed. In the fourth section, a feed flash unit was modelled to calculate the bubble-point temperature of the feed, given its composition and pressure. Homotopy-continuation method has been built into this system to solve the necessary system of equations. In the final section the actual distillation model equations are represented, which could then be executed by restarting from the processed information from the fourth segment. By adopting this modular representation of the entire system. the only change that is required for solving similar distillation optimization problems is modifying the physical property database. All other sections remain intact. For more information on this systematic development, a GAMS representation of a distillation optimization problem can be found in Vasantharajan (1989).

6.2.2. Optimization of Rigorous Distillation Columns

The first of the two nonideal example distillation optimization problems solved is due to Naka *et al.* (1979) and is an acetone-acetonitrile separation system, equipped with a partial condenser. The source for the second distillation problem is Gallun and Holland (1976), whose objective is to separate methanol and chloroform. Unlike Lucia *et al.*, who used a partial condenser, we decided to preserve the original intent of the authors, and use a total condenser. For these problems, the column related variables are the liquid and vapor mole fractions, the liquid and vapor flowrates, the temperatures and the condenser and reboiler heat duties (for details refer to Vasantharajan, 1989). The objective function used here to represent the operating conditions is:

$$Mm \quad \bullet = 3.0 \times 10^{"7} \ (Q^{R} - Q^{C}) \ - \ (LK^{C} \bullet HK^{R}) \qquad (9)$$

Here, the superscripts C and R denote the condenser and reboiler, respectively. LK and HK the light and heavy key component flow rates. Q denotes the heat duties. We have adopted here the objective function form used by Lucia and Kumar.

The optimization results in Table 2 bear testimony to the exemplary robustness and efficiency of the reduced SQP strategy. Of the six cases considered, version (5.1) of MINOS failed on three instances. Version (5.2) succeeded in solving one of these problems, but was unable to tackle two cases of *Distillation Nonideal (2K* which are by far the toughest cases solved here. For this problem, the initial set of values used for the variables were selected in a relatively arbitrary fashion by Gallun and Holland, so that they could not be regarded as good first guesses in the sense of being close to the solution. Thus, simulation of the column, or its simultaneous optimization from this starting point is not easy; these are the two cases on which MINOS failed. RND, on the other hand, was highly successful in solving all six cases. Further, on the cases in which both algorithms reached the optimal solution, RND was considerably more efficient than MINOS, and, in fact, required an order of magnitude lower number of model evaluations.

The initial and final values for the objective function and critical variables for both these problems are given in Vasantharajan (1989). Lucia and Kumar reported the optimal values for the acetone-acetonitrile splitter case alone, and these are given there for comparison. In addition, they discuss in the separation details of this particular example. As can be seen from the results for this problem, there is a small difference in the final values of the variables and the objective. In their formulation, the column was operated isobarically, and the vapor phase was assumed to be ideal. In the current formulation, however, a rigorous model for the vapor phase fugacity and enthalpy is used, and a linear pressure variation across the column is permitted, making the column representation more rigorous and complex. This extra rigor in modelling will explain the marginal differences in the optimal solutions.

For the starting values given in Gallun and Holland, the methanol-chloroform separation is very poor. About 63% of the methanol in the feed is taken overhead, with only 20% purity of the distillate product. Likewise, about 62% of the heavy key, chloroform is separated in the bottoms, with a purity of 20%, indicating tremendous scope for improvement. The simulation results obtained from this starting point are presented in Vasantharajan (1989), and demonstrate a much improved separation of these key components. Approximately 94% of the methanol is now carried overhead at a purity of about 36%, while an equal percentage of chloroform in the feed is recovered in the bottom stream, at a purity of about 30%. However, considerable increase in the condenser and reboiler duties is required to effect this sharp separation. The new utility consumption represent a 225% increase over the specified initial values.

The main improvement in the operating conditions of this column at the optimal solution is in the reduction of heat duty requirements. At the solution, the optimal utility requirements represent a modest 12% increase over their initial values. In addition, 99.8% of the methanol is recovered in the overhead product stream indicating an overall improvement in the column performance. This is at the expense of a small decrease in the chloroform recovery in the bottoms. A comparison of the initial and optimal values for this column reveals a significantly better separation performance with an infinitesimal increase in the utility requirements. Identical results are exhibited by a simultaneous approach to optimize the column, confirming the viability of such techniques to solve complex separation problems with the reduced SQP approach.

7. Conclusions

In this study the range and null space decomposition strategy (RND) has been extended to solve large-scale nonlinear programs. A reduced SQP implementation using sparse matrix processing techniques has been developed using this strategy, and interfaced with an efficient modelling front-end, GAMS. In addition, systematic procedures for tackling problems associated with QP-based methods like inconsistent linearizations or infeasible quadratic programs have been included. Facilities for automatic generation of the required nonsingular basis has been developed and integrated in this package.

The reduced SQP has been extensively tested against MINOS on a variety of general nonlinear problems, and a series of process optimization problems. The problems solved range from 34 variables and 23 constraints to 977 variables and 975 constraints. Numerical comparison based on these diverse problems demonstrates the efficiency and robustness of the RND strategy to solve difficult and complex models. MINOS, on the other hand, was unable to solve two cases of the largest problem considered here. Further, the results show that both in terms of the number of function evaluations required, and in terms of the computational effort required, RND is almost always superior to MINOS.

In addition, this study describes a new equation-based representation of distillation based models. Based on a systematic, conceptual decomposition of the tasks involved in modelling separation processes rigorously, and exploiting the facilities in the modelling environment, GAMS, a highly modular and flexible depiction of general distillation columns has been achieved. Using the reduced SQP implementation, the operating conditions of these models have been optimized, using both sequential and simultaneous techniques to solve them. The results confirm the efficacy and the viability of the decomposition strategy, RND, to solve large, arduous problems. Further, contrary to published results, they show that the simultaneous paradigm is an efficient alternative to solve these difficult problems.

Finally, improvements of the current implementation have been planned, in terms of incorporating more efficient LP and QP solution techniques, and better sparse matrix solvers. For the QP subproblems, a new and promising large-scale technique, PDQP by Ng and Thompson (1986), has yet to be tested. Preliminary results obtained in solving random quadratic programs have demonstrated tremendous potential of this technique. Efforts to integrate this strategy in the reduced SQP implementation are currently underway.

REFERENCES

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

Brooke, A., D. Kendrick and A. Meeraus, -GAMS, A User's Guide," Scientific Press, Redwood City, CA (1988)

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).

Edgar. T.F. and Himmelblau. D. M., *Chemical Process Optimization.* McGraw-Hill, New York (1988).

Gallun. S.E. and CD. Holland. "Solve more Distillation Problems," *Hydrocarbon Processing,* 55, pp. 137-144 (1976)

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

Kumar, A. and A. Lucia, "Separation Process Optimization Calculations," *IMACS J. AppL Num. Math.*, 3, p. 409-425 (1987)

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

Lucia, A. and A. Kumar, "Distillation Optimization," *Comp. and Chem. Engr..* 12, 12, pp. 1263-1266 (1988)

Murtagh, B.A. and M. A. Saunders, "Minos/Augmented Supplementary User's Manual", Report OR/78 /6, The University of'New South Wales, Australia (1978).

Naka, Y., M. Araki and T. Takamatsu, "A new procedure for calculating composition profiles in distillation by a matrix method," *Int. Chem. Eng.%* 19, pp. 137-145 (1979)

Ng, S-M and G.L. Thompson, -A Primal-Dual Dimension Expanding Quadratic Programming Algorithm," GSIA Technical Report (1986)

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

Prausnitz, J.M., T-F. Anderson, E. Grens, C Eckert, P. Hsieh and J. P. O'Connell, *Computer Calculations for Multicomponent Vapor-Liquid and Liquid-Liquid Equilibria.* Prentice-Hall, Englewood Cliffs, NJ (1980)

Reklaitis, G.V., A. Ravindran and R. Ragsdell, *Engineering Optimization*, Wiley, New York (1983)

Vasantharajan, S., PhD Thesis, DepL of Chemical Eng., Carnegie-Mellon Univ., Pittsburgh, PA (1989).

Vasantharajan, S. and L.T. Biegler, "Large-Scale Decomposition Strategies for Successive Quadratic Programming," *Computers and Chemical Engineering*, X2. 11, p. 1087 (1988).