A Robust Technique for Process Flowsheet Optimization Using Simplified Model Approximations

by

N. Ganesh and L. Biegler

EDRC-06-28-87 >

A ROBUST TECHNIQUE FOR PROCESS FLOWSHEET OPTIMIZATION USING SIMPLIFIED MODEL APPROXIMATIONS

by

N. Ganesh and L. T. Biegler[1]

Chemical Engineering Department

Carnegie-Mellon University

Pittsburgh, PA 15213

Submitted to R. R. Hughes Memorial Issue of Computers and Chemical Engineering

January, 1987

[1] Author to whom correspondence should be addressed

ABSTRACT

Recently, embedded simplified process models have been shown to be very efficient for process simulation. When compared to the direct use of rigorous models, this approach has the potential to reduce the computational effort of process simulation by up to an order of magnitude or more. Application of this approach to process optimization should therefore lead to similar savings in computational effort as well as substantial improvement of the process.

However, current simplified model embedding schemes applied to process optimization cannot, in general, converge to the optimum defined by the more rigorous process models. Consequently, they require an expensive rigorous model optimization starting from the solution of the simplified model optimum to guarantee convergence.

In this paper we develop a framework that incorporates simplified models into an optimization algorithm *and* guarantees convergence to the rigorous model optimum. Here rigorous process models are evaluated only when necessary to insure progress toward the optimal solution. A theoretical justification of the algorithm is presented and several process examples are solved to demonstrate the effectiveness of this approach.

SCOPE

Modeling is an integral part* of process design and has a direct influence on solution of flowsheet simulation and optimization problems. Here, in order to represent accurately the solution of a simulation or optimization study, complex, nonlinear models are required to describe the system behavior. These large-scale nonlinear models coupled with the iterative nature of the solution procedure make process simulation and optimization computationally intensive. For thermodynamic models, for example, it has been estimated that physical property evaluations require between 75% to 90% of the computational overhead, depending on the complexity of the modeling subroutines used [13,19].

Recently, several researchers have developed efficient calculation strategies with the aim of alleviating the computational problems associated with process simulation and optimization. In particular, the application of local simplified models has received a great deal of attention [1,8,9]. Compared to the time demands of more complex thermodynamic model computations, some studies have reported reductions in computing time requirements on the order of 70 to 85% by using appropriate simplification schemes for physical property evaluations. In most approximation schemes, the simplified models are constructed by deriving reduced linear or nonlinear approximation functions that approximate a specified rigorous model over local regions.

The success of the local model approximation approach in saving time for simulation has promoted several researchers to extend this technique to process optimization [9,11,18]. In this case, the simplified models not only offer savings in function evaluations but also aid in gradient calculations. Reduced model equations often have analytic derivatives, so gradient information can be obtained with relatively little computational effort In the "inside-out" optimization strategy [6,17], as shown in Figure 1, rigorous models are retained at the outer level of the two-tier model configuration. A set of reduced analytic models is introduced at the inner level of this two-tier model hierarchy. The model at the outer level is used to fit parameters for the simplified models at the inner level. Using flash problems and an ammonia synthesis process simulated on FLOWTRAN, Jirapongphan et al [18] reported remarkably short optimization times (on the order of one to five simulation time equivalents). Recently, this concept has been extended further to include reduced models for process units such as absorbers and reactors within the ASPEN PLUS Simulator [11]. Again remarkably short computation times were observed for optimization.

None of these simplified model approaches, however, guarantee convergence to the optimum of the original problem with rigorous models. In fact, they generally converge to suboptimai solutions. In this paper we develop a framework for simplified model embedded optimization that *does* guarantee convergence to the optimum of the original problem. We further demonstrate its efficiency and performance in a conservative manner on three flowsheet optimization problems.

CONCLUSIONS AND SIGNIFICANCE

A new framework has been developed for embedding simplified models within flowsheet optimization problems that contain complex and rigorous process models. Based on generating search directions that are also descent directions for the rigorous model exact penalty function, this approach guarantees convergence to the optimal solution of the rigorous process model.

The approach is first developed by considering the theoretical properties of a one-parameter exact penalty function. These serve as a justification for deriving a simple procedure for screening appropriate search directions based on simplified models. An approach centered on the Successive Quadratic Programming (SQP) algorithm is then described and its steps are illustrated with the help of a small, analytic optimization problem.

The new rigorous-simplified model (R/S) algorithm is next applied to three comprehensive flowsheet optimization problems. No effort was made to tailor the simplified process models to specific flowsheet examples, so these process optimizations represent a rather conservative test for the efficiency of the algorithm. Even so, with the R/S algorithm savings in CPU time were observed on all three process problems. On all but the simplest problem the R/S algorithm also outperformed the two-stage restart procedure, where the rigorous model is optimized from the simplified model optimum. Moreover, the R/S algorithm converges to the rigorous model solution even under poor approximation schemes.

Lastly, it should be noted that development of this approach is independent of the choice of simplified models considered for process optimization. Clearly this choice is often problem dependent although important guidelines for model selection are available in [8, 25]. As shown below, and as expected intuitively, performance of the R/S algorithm improves with better pointwise agreement of simplified and rigorous models, as well as with the availability of analytic gradients for simplified models.

1. APPROXIMATION AND OPTIMAL SOLUTION

Flowsheet optimization deals with minimization (or maximization) of a cost based objective function subject to satisfaction of the simulation model and any additional process related constraints. The mathematical formulation can be given as:

$$\min \ f(x,y)$$
$$\text{s.t. } g(x,y) \leq 0 \qquad (1)$$
$$h(x,y) = 0$$

where:

ϕ - objective function

g - design inequality constraint

h - equality (tear) constraint

x - decision variables

y - tear variables

Note that in this general formulation we have explicitly included the set of equality constraints, h = 0. Here the flowsheet equations that correspond to the convergence of tear streams have been specified as a set of equality constraints, i.e. a reduced modular form of the simulation problem has been incorporated within the optimization problem. The inequality contraints, $g \leq 0$, correspond to the set of design constraints that may arise from operational restrictions, e.g. bounds on reactor temperature, column pressure, product purity. The necessary termination criteria for the optimization problem are given by first order Karush-Kuhn-Tucker (KKT) conditions:

$$\nabla \phi(x,y) + \nabla g(x,y)\bar{u} + \nabla h(x,y)\bar{v} = 0$$

$$\bar{u}^{*}g(x,y) = 0 \; ; \quad V \geq 0 \; ; \quad \hat{}(x.y) \; \hat{} \; 0 \qquad (2)$$

$$h(x,y) = 0$$

where

$\bar{u},\bar{v}$ - KKT multipliers of g,h respectively

For the later development in this paper, the flowsheet is considered optimized when these optimality conditions are satisfied to a preset tolerance. An important feature of the optimality conditions is that they depend not only on function values but also on the gradients of the constraint and objective functions with respect to the process variables x and y.

In optimizing a flowsheet in sequential modular fashion, process correlations and physical property equations associated with a process module are solved internally within the model subroutine. The complexity of these modeling equations within the individual modules determines the complex or "rigorous" nature associated with the overall flowsheet model. While use of more simplified or approximate models is therefore effective for simulation, the fundamental disadvantage of the model approximation procedure for flowsheet optimization is:

The candidate optimum generated by using the simplified model embedding scheme may prove to be suboptimal, because it may not satisfy Karush-Kuhn-Tucker iKKTj conditions of the original optimization problem, defined by the corresponding rigorous modeling equations.

When a simplified model approximation procedure is said to solve the optimization problem from an initial guess of decision and tear variables, the approximation affects not only the function values but also the gradients at any iteration. It is important to note that the local model approximations are developed by comparing or simply matching function values between the rigorous and simplified models. However, at the same point, the *gradients* computed by the simplified model may differ significantly from rigorous model derivatives. This is due to a structural change in the model that accompanies the simplification procedure.

In solving the optimization problem gradients play a key role in determining the search direction from a current base point to the next Moreover, convergence to an optimal solution is ascertained by checking the optimality (KKT) conditions, which require accurate gradients. With a simplified model embedding scheme, KKT conditions are satisfied for the *simplified* model with parameters selected so that *function* values for simple and rigorous models match. However, since their gradients generally will not match, this point is usually suboptimal for the original problem. Biegler [2] illustrated the problem of convergence to a false optimum with an analytic example. Later, it was shown that in some cases the model simplification scheme may lead to convergence failures even though the original problem has an optimum [5]. Trevino-Lozano [11,25] also discussed this problem and, as a remedy, proposed a two-stage restart procedure that involves re-solving the optimization problem from the optimal solution of the simplified model problem, using only rigorous models.

Instead of the restart procedure, we develop in the next section a theoretical framework that ensures convergence of an embedded simplified model optimization problem to the optimal solution for the original, rigorous model. We note that this scheme is independent of the simplified models chosen for embedding. Although we do not explore any guidelines for choosing appropriate simplified models, (These can be found elsewhere [8,25]) we present an algorithm that utilizes this framework and study its performance with the help of analytic and process flowsheet examples.

2. DEVELOPMENT OF A THEORETICAL FRAMEWORK

In developing **a** framework that ensures convergence to the optimal solution of the rigorous model problem, we postulate that the optimization algorithm can move from a current base point to the next base point only if we can determine that the next point is an improved guess for the rigorous model optimum. In order to satisfy this requirement, we use a one-parameter *exact penalty function* evaluated in the space of the rigorous model, as a quantitative measure to guide the use of a search direction. The reasoning for using the exact penalty function arises from the following properties:

Property 1.

Let $z^* = [x^{*T}\ y^{*T}]$ be a local minimum for problem (1) which satisfies the first order KKT conditions as well as the second order sufficiency conditions (see [15]), then for any vector norm and with Q satisfying:

$$Q(O > 0 \text{ for } C > 0 \text{ and } \lim_{\sigma \to 0^+} \frac{dQU}{dC} * 0$$

as well as:

$$a > a = \frac{\| \bar{u}.\bar{v} \|'}{\frac{dQ}{d\zeta}(0^+)}$$

the point z* is a strict local minimum for the exact penalty function:

$$P^Q(z,*) = <f)(x) + a\ Q(\| g_+(z), h(z) \|)$$

where $C_+ = \max(0,C)$

Moreover, any local minimum of this penalty function satisfies first *and* second order KKT conditions. The proof of this property is given in Han and Mangasarian [15].

If we choose the 1-norm and let Q(C) = C we have a simple one-parameter penalty function given by:

$$P(z,\alpha) = \phi + \alpha \left[\sum_{j=1}^{n} |h_j| + \sum_{i=1}^{m} \max(0, g_i) \right]$$

Note **that the above property holds** for *any* α above the threshhold value. Also, note that although these penalty functions are not everywhere differentiate, their directional derivatives,

$$D_d P^Q = \lim_{\epsilon \to 0} (P^Q(z + \epsilon d) - P^Q(z))/\epsilon$$

do exist everywhere (see [14]). Therefore, minimization of an exact penalty function based on the function values of the rigorous model will solve the nonlinear program (NLP) of problem (1).

Satisfaction of the optimality conditions (Eq. 2) can be achieved through application of the Successive Quadratic Programming (SQP) algorithm to the rigorous model optimization problem. Here the following quadratic program is constructed and solved at each iteration:

$$\min_d \; \nabla\phi(z^i)^T d + \frac{1}{2} d^T B^i d$$

$$\text{s.t.} \quad g(z^i) + \nabla g^T(z^i)\, d \le 0$$

$$h(z^i) + \nabla h^T(z^i)\, d = 0 \qquad (3)$$

where

z = process variable, $z \equiv \begin{Bmatrix} x \\ y \end{Bmatrix}$

B = Hessian matrix or its approximation

The results of the solution at each iteration give a search direction, d, in (x, y) as well as shadow price vectors, u and v for the constraints g and h respectively.

Property 2

If problem (3) has a solution with bounded multipliers u and v and B is positive definite, then $d \neq 0$ will give a descent direction for the exact penalty function:

$$P(z,\alpha) = \phi + \alpha \left[\sum_{j=1}^{n} |h_j| + \sum_{i=1}^{m} max\ (0,\ g_i) \right]$$

with

$$\alpha > max \left(u_i, |v_j| \right) \qquad (4)$$

The proof of this property is a generalization of Theorem 3.1 in Han [14] and is shown in the Appendix.

Once we have a (nonzero) search direction, d, from solving eqn (3), a stepsize, λ, along this direction is guaranteed to exist for the next point

$$\begin{Bmatrix} x^{i+1} \\ y^{i+1} \end{Bmatrix} = \begin{Bmatrix} x^{i} \\ y^{i} \end{Bmatrix} + \lambda\ d \qquad (5)$$

Han [14] showed that if the stepsize, λ, is chosen by reducing the exact penalty function along the search direction, d, then the (SQP) algorithm converges to a KKT point from any starting point. Therefore, the exact penalty function (Eq. 4) is suitable for forcing *global convergence*. It should be noted that the exact penalty function has been used as a merit function for several optimization algorithms [10,23].

Current simplified model embedding schemes based on SQP derive QP subproblems that use function and gradient values from the *simplified* model. This QP is, of course, inconsistent with the original rigorous model-based optimization problem and its solution d_s, may *not* be a descent direction for the original problem. Fortunately, we can ascertain if such a direction, d_s will progress toward the rigorous model optimum, before moving along this direction from the current point We propose the following strategy to accomplish this.

- Select the one-parameter exact penalty function, P in Eq. 4, based on rigorous function evaluations, as the merit function to be reduced in the direction d_s.
- Determine the directional derivative, $D_{d_s}P$, of this penalty function along this direction.
- If $D_{d_s}P < 0$, carry out a line search to determine the stepsize, λ, along d_s.
- If $D_{d_s}P \geq 0$, abandon the search direction, d_s; evaluate the rigorous model gradients at the point and solve the rigorous QP to compute a new search direction, d_r.
- Carry out a line search along d_r to determine the stepsize, λ.

This method ensures that the simplified model search direction, d_s, moves toward the optimum only if an improvement is achieved for the rigorous exact penalty function. Since an exact penalty minimization ensures global convergence [15], the procedure guarantees that the next base point is indeed a better point for the rigorous model optimization. Note that this procedure is not limited to d_s generated by the SQP algorithm or the choice of any particular simplified model. The above procedure can be adapted equally well to any of the simplified model embedding schemes cited above. For example, with Jirapongphan's inside-out strategy [17] (see Fig. 1), the search direction could result from solution of the inner (simplified) minimization problem; progress could be ascertained by then comparing the exact penalty function at successive solutions.

Used in SQP, however, it is well known that the exact penalty line search function may lead to small stepsizes and slow convergence rates in the neighborhood of the optimal solution

[21]. As a result, many investigators have proposed other line search strategies to relax the stepsize procedure. These include the Powell line search function [23], the "watchdog" technique [7] and augmented Lagrangian functions [3,24]. In a number of computational studies, these functions used with SQP have superior performance characteristics.

All of these functions, however, require values of the multipliers from the QP solution and when d_s is computed these values are generally not accurate. Moreover, we do not expect to calculate d_s in the neighborhood of the optimal solution. Consequently, we have chosen the one-parameter penalty function, since its only requirement is that the parameter a be sufficiently large. Moreover, when a simplified model QP solution is *not* a descent direction for the exact penalty function, the rigorous model QP can be solved to get a new search direction, d_r. For these QP solutions we now apply superior line search functions (such as a modified augmented Lagrangian function) along d_r and thus obtain larger stepsizes along d_r to improve the overall efficiency of the optimization procedure.

3. ALGORITHM FOR RIGOROUS-SIMPLIFIED MODEL OPTIMIZATION

Based on the previous section, we now formally state an algorithm that can be applied toward a general rigorous-simplified model framework. The procedure presented here is specific to the SQP algorithm *without* inner optimization loops although, as mentioned above, it can easily be modified to handle other approaches as well.

1. Given an optimization problem, identify the rigorous models that need to be approximated.

2. Choose an appropriate simplified model scheme to approximate the selected set of rigorous models. (Guidelines for choosing these models are given in [8^25]and may include local parameter fitting, or matching of function values at a given point.)

3. Initialize the decision variables, x_i and tear variables, y_i and set the iteration counter $i = 1$.

4. Compute the objective function, ϕ, and constraints g and h based on the rigorous models at the base point

5. Compute the function values, ϕ, g and h based on the simplified models at the same point Evaluate the simplified model gradients analytically or by perturbation.

6. If $i = 1$, set Hessian approximation B^i to some suitable positive definite matrix (usually, I). If $i > 1$, update B^i using the BFGS update.

7. Solve the following simplified model based quadratic program, where the subscript, s, refers to simplified model values:

$$Q_s(z^i, B^i): \quad \min_{d_s} \quad \nabla\phi^T(z^i)\, d_s + \tfrac{1}{2} d_s^T B^i d_s$$

$$\text{s.t.} \quad g(z^i) + \nabla g^T(z^i)\, d_s \le 0$$

$$h(z^i) + \nabla h^T(z^i)\, d_s = 0 \qquad (6)$$

where

$$z = \text{process variable}, \quad z = \begin{pmatrix} x \\ y \end{pmatrix}$$

8. Set α such that $\alpha > max\,(\alpha, |v_i|)$ where u and v are the shadow prices from the QP in step 7.

9. Along the search direction, d_s evaluate the directional derivative of the rigorous exact penalty function by perturbation:

$$D_{d_s} P = \frac{P(z^i + \varepsilon\, d_s) - P(z^i)}{\varepsilon} \qquad (7)$$

where

ε = perturbation factor for search direction, d_s

10. If $D_s P < 0$, carry out a line search along d using the exact penalty to find a stepsize, X_s, that yields an improved point Each point requires calculation of the rigorous functions:

$$\phi(z),\ g(z\backslash\ h\{z)$$

where

$$z \equiv \begin{Bmatrix} x^i \\ y^i \end{Bmatrix} + \lambda_s d_s$$

11. If $P(z + \backslash d)_s < P(z') + 0.1XD_{d_s} P$, set

$$if) - \begin{Bmatrix} x^i \\ y^i \end{Bmatrix} + \lambda_s d_s$$

and / = / + 1; return to step 4.

12. If $D_{d_s} P \geq 0$, then compute the rigorous model gradients at $\wedge x\backslash\ y^x)$

13. Set up the rigorous model based QP and solve the QP to get the search direction, d_r where subscript, r, refers to search direction from a rigorous model based quadratic program.

14. Compute the KKT error to see if the KKT conditions are satisfied to within a specified tolerance at the current point. If KKT error < €, STOP.

15. Carry out a line search along d_r using the augmented Lagrangian strategy [3] to determine the stepsize, X_r .

16. Set

$$\begin{Bmatrix} x^{i+1} \\ y^{i+1} \end{Bmatrix} = \begin{Bmatrix} x^i \\ y^i \end{Bmatrix} + \lambda_r d_r$$

and $l = l + 1$; return to step 4.

Having stated a rigorous-simplified (R/S) optimization algorithm for SQP_f we now illustrate its operation with a small analytic example.

4. ANALYTIC EXAMPLE FOR RIGOROUS-SIMPLIFIED MODEL ALGORITHM

Consider the problem:

$$
\begin{aligned}
\min_{x,y} \quad & \phi : -x^2 - y^2 \\
\text{s.t.} \quad & g : x + y - 1 \le 0 \\
& h^r : y - x^2 - x + 0.6 = 0 \\
& h^s : y - x^2 = 0 \\
& y \ge 0 \; ; \; x \ge 0
\end{aligned}
$$

Here h_r is the rigorous model equality constraint and h_s its simplified equivalent. The optimal solution to the rigorous model problem lies at Point A in Fig. 2,

$$(x^*, y^*) \cong (0.6125, 0.3876)$$

and the corresponding objective function value is $\phi^* = 0.5253$.

The simplified model optimum for this problem is at Point B in Fig. 2:

$$(x^\circ, y^\circ) \cong (0.6180, 0.38197)$$

and the objective function at this point is $\phi^\circ = 0.5279$.

The derivatives of the objective function with respect to x and y are given by:

$$\left[\ \partial\phi/\partial x\ ,\ \partial\phi/\partial y\ \right] = \left[\ -2x\ ,\ -2y\ \right]$$

The gradients of the rigorous model based equality constraint are given by

$$\left[\ \partial\eta/\partial x\ ,\ \partial\eta/\partial y\ \right] = \left[\ -2x - 1\ ,\ 1\ \right]$$

and the simplified model based equality constraint gradients are

$$\left[\ \partial h/\partial x\ ,\ \partial h/\partial y\ \right] = \left[\ -2x\ ,\ 1\ \right]$$

Let us assume an initial guess **(Point C) at** $(x^0, y^0) = (2, 2)$. **Based on** the simplified **functions and** gradients at this point, and with **the Hessian** matrix set to identity, the following **quadratic program can be formulated:**

$$\begin{bmatrix} 1 & 0 & 1 & -4 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ -4 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ u \\ v \end{bmatrix} = - \begin{bmatrix} -4 \\ -4 \\ 3 \\ -2 \end{bmatrix}$$

Note that the linearized inequality **constraint,** $g \leq 0$ **has been included in** the **QP.** The solution to the quadratic program yields a search direction, d:

$$d_s \equiv (\Delta x \ , \ \Delta y\ ) = (\ -1\ ,\ -2\ )$$

and $$(\ u\ ,\ v\ ) \equiv (\ 5.8\ ,\ 0.2\ ).$$

In order to determine whether d_s is a descent direction for the rigorous exact penalty function, first set

$$\propto\ >\ \max\ (\ u,\ |\ v\ |\ )$$

Let $\propto\ = 10$. The rigorous model based penalty function value at (2,2) can be evaluated from:

$$P\ (x,y,\propto) = \phi + \propto \left(\sum \left(g_+ + |\ h_r\ | \right) \right)$$
$$P\ (2,2) = (-8) + 10\ (\ 3 + 3.4\ ) = 56$$

Introduce a perturbation in d_s using a perturbation factor, say $\xi\ =\ 0.001$. This gives the perturbed point at:

$$(\ x^r\ ,\ y^r\ ) = (\ 1.999\ ,\ 1.998\ )$$

Evaluate the rigorous penalty at (x^r,y^r) so that

$$P\ (x^r,\ y^r) = -7.988005 + 10\ (2.997 + 3.397001) = 55.952$$

Compute the directional derivative along d_s from:

$$(\ P\ (1.999,1.998) - P\ (2,2)\ )\ /\ \xi = -47.995 < 0$$

Since the direction d_s from the simplified QP gives a descent direction for the exact penalty, perform a line search along d_s. Here we initially set $X_s = 1$ (full step). This gives:

$$(x^{\circ} + X_s AX \ . \ y^{\circ} + X_s Ay) = (1.0)$$

Compute the exact penalty at (1,0) :

$$P\ (1,0) = (-1) + 10\ (0 + 1.4) = 13$$

Since the value of the exact penalty function at the full step is less than the exact penalty at the base point (2,2), this represents an acceptable move. The improved guess for rigorous optimum becomes:

$$(x^1, y^1) = <1.\ 0\ )$$

Note that although the direction was provided by the simplified model QP, it acts as a descent direction for the rigorous model problem also. This can be seen from Fig. 1. Here, the starting point (2,2) (point C) is sufficiently far from both the simplified and rigorous optimal solutions so that the simplified model QP performs effectively in finding a descent direction for the rigorous model problem also.

When applied iteratively to this problem the rigorous-simplified (R/S) algorithm converges to the correct solution. Note that as the algorithm proceeds toward the optimum, a rigorous search direction will eventually be required. As seen above, the search direction found by solving the simplified model QP at point C provides a descent direction toward the rigorous optimum; hence we can move from C along this direction. On the other hand, at point D in Fig. 2, a search direction, $d^D_{s,t}$ obtained from the simplified QP will not will not move toward satisfying the rigorous constraint, /7^ng. In this case, $VP^T d_s^{\circledR} > 0$ and we need to evaluate a new direction at D based on the rigorous model QP. In fact, if we start anywhere between points A and B, a move based on the simplified model constraint will obviously not improve the rigorous model constraint Here simplified model QP solutions cannot aid convergence but merely require more function evaluations.

For simplicity this analytic example does not include adjustable parameters in the simplified model. While adding these would change the location of point B, we would still generally observe a rejection of the simplified model search direction in the neighborhood of point A. Depending on how well the simplified and rigorous models agree, this rejection could occur arbitrarily close to point A. (Conversely, the simplified QP solution could be rejected at each iteration if model agreement were very poor.) In fact, even if both optima were to coincide, the above algorithm still applies. Here the simplified search direction at the optimum and the corresponding directional derivative would be zero. At this point the R/S algorithm would reject the search direction, solve the rigorous model QP and thus confirm (also with a zero search direction) that the point was indeed optimal.

5. PROCESS FLOWSHEET EXAMPLES

The analytical example in the previous section illustrates some of the properties of the R/S algorithm. Here we consider how the algorithm performs on actual flowsheet optimization problems. At each iteration the computational effort for the rigorous model optimization alone is given by:

$$(n \times f + n + 1) \times FE^x$$

where:

n_d - number of decision variables

n_t - number of tear variables

f - fractional flowsheet perturbation for decision variables

FE^r- CPU time for a rigorous model flowsheet pass.

On the other hand, the effort for the R/S algorithm at each iteration is given by:

$$(n_d \times f + n_t) \times (FE^s + f \times FE^r) + \psi (FE^s + 2 \times FE^r)$$

where:

FE^S- CPU time for a simplified model flowsheet pass

$$\psi = \begin{cases} 0 \text{ if } d_s \text{ is a descent direction} \\ 1 \text{ if } d_s \text{ is not a descent direction} \end{cases}$$

Since FE^s is normally much smaller than FE^r (and negligible if analytic gradients are available) the main additional cost, if the search direction is rejected, is an additional rigorous model flowsheet perturbation per iteration. On the other hand, as seen from the above equations, considerable savings are obtained if d_s *is* a descent direction. As discussed above, acceptance or rejection of d_s depends on how well simplified and rigorous models agree.

We now consider three process flowsheet examples to study the performance of this algorithm. These examples present a rather conservative test of the algorithm because:

- simplified and rigorous models were considered by interchanging physical property options without adding adjustable parameters to tailor the simplified models

- gradients for simplified models were evaluated by flowsheet perturbation

Consequently, we have not "optimized" the simplified models to the specific application. Guidelines for doing this are available in [8,25], but choosing appropriate simplified models is normally a problem specific task. Instead, we are mainly interested in whether even crude simplified models have the potential for time savings, and whether convergence to the rigorous model optimum occurs with the R/S algorithm.

Simple Flash Recycle Flowsheet

A simple flash recycle problem flowsheet is used to demonstrate the application of the R/S. optimization algorithm. The flowsheet is shown in Fig. 3 and was originally presented in Biegler and Hughes [4]. Here a light hydrocarbon feed is mixed with recycled bottoms and flashed adiabatically. Vapor is removed as product and the liquid is split into a bottoms product and the recycle, which is pumped back to the feed. Problem specifications are presented in Table. 1. The process was optimized using the SPAD (Simulator for Process Analysis and Design) simulator developed by Hughes and coworkers [16].

The flowsheet includes two decision variables, the splitter ratio and the pressure in the flash. The six component flowrates and the specific enthalpy of the recycle stream make up the seven tear variables for the problem. Since the outlet pressure of the pump is fixed in this case, the recycle stream pressure need not belong to the set of tear variables. Two different objective functions are considered for this flowsheet. As seen in Table 1, the objective function for the *monotonic* optimization problem corresponds to the flowrate of the lightest component in the flash overhead; for the *nonlinear* case a complex combination of the component flows in the flash overhead was maximized.

The simplification scheme was- implemented at the level of thermodynamic models and applies to the vapor-liquid equilibrium calculations in the flash unit as well as specific enthalpy correlations in the process streams. The rigorous model for the hydrocarbon system is represented by the Soave-Redlich-Kwong **(SRK)** option in **SPAD.** The ideal option based on Raoult's law, Antoine's equation and ideal enthalpies serves as the system's simplified counterpart Implementation of the R/S algorithm required extensive modifications to the executive of the SPAD simulator but did not affect the existing modular structure of SPAD. Optimization studies were carried out on a DEC-20 computer at Carnegie-Mellon University.

Four different cases were compared in solving both the monotonic and nonlinear optimization problems. Here we considered the fully rigorous model solution, a solution based entirely on the simplified model, the R/S algorithm and an approach based on the restart method. The restart method is similar to the method proposed by Trevino-Lozano [.253. Here rigorous models are used exclusively to reoptimize the problem from the solution obtained from the simplified models. For this example, the rigorous model solution identifies the true optimum as the reference solution. The simplified model solution indicates how accurately the approximating model follows the behavior of the rigorous model and the relative speed with which the simplified model can be evaluated. The R/S algorithm solution illustrates:

- the ability of the method to converge to the true optimum
- the improvement in performance with respect to reducing the number of rigorous model steps and the corresponding savings in CPU time.

The results of the monotonic flash optimization are given in Table. 9. In this case, both the SRK and Raoult's Law models converge to the same optimal decision variable values, although values for the tear variables and the objective function differ slightly. The RS method not only converges to the rigorous model optimum but also gives the correct value of the objective function evaluated from the rigorous models. Here the number of rigorous model steps is 5 while the complete rigorous model solution requires 8 iterations. The R/S algorithm achieves this by using 2 simplified model steps *en route* to the solution. For these directions full steps are taken. Thus accelerated convergence is effected and the overall time required for optimization decreases from 184 CPU sees, corresponding to the rigorous case, to 135 CPU sees, a reduction of 26.6%. In this case, as in all other cases with SPAD, the simplified model gradients were evaluated from flowsheet perturbations. If we compare the time savings on the basis of the number of rigorous gradient based iterations, then a reduction of 3 rigorous steps corresponds to a time saving of 37.5% over the complete rigorous solution. Here the restart method gives the best time reduction of all the methods for this problem (79 sees). This occurs because the number of iterations required by the rigorous model in the second stage (4) is half that required by the rigorous procedure starting from the initial point (0.5, 40). Also, note that these iterations are required even though the decision variables have the same optimal values.

For the nonlinear objective function, the simplified model solution is different from the rigorous model optimum with respect to flash pressure and the tear variables. This also leads to a 33% difference in the objective function value (Table 3). This example demonstrates that the simplified model optimization generally leads to a different optimal solution especially if the objective function is highly nonlinear. Unlike the simplified model case, the R/S algorithm successfully locates the optimal solution for the rigorous model. Here the number of rigorous steps is 7 whereas the complete rigorous model solution requires 11 iterations. This represents a 36.4% reduction in the time spent in the evaluation of rigorous model based flowsheet. The actual time taken by the R/S algorithm is 184 CPU sees, which corresponds to a 20.7%

reduction over the 232 CPU secs required by the rigorous solution.

For this problem, a restart from the simplified model optimum requires more overall time (193 secs) than is required by the R/S algorithm. Here one sees that with significantly different optimal solutions, the restart method is not necessarily an efficient procedure.

The two flash examples demonstrate improved performance of the R/S algorithm over the corresponding rigorous model based optimization. The primary goal of ensuring convergence to the correct solution is also achieved and the method shows the potential to register a reduction in CPU time. Again, it should be mentioned that the actual savings for these two problems could have been higher had the simplified gradients been computed analytically. Thus the above results can be considered as a worst case estimate of the performance that can be achieved with this optimization strategy. In the next example we consider a more complex optimization problem where the agreement between the rigorous and simplified models is poor. This problem therefore represents a severe test of the R/S strategy.

Ethylbenzene Flowsheet Problem

The flowsheet shown in Fig. 4 represents a simplified version of the ALKAR process (see [20] used in the manufacture of ethylbenzene from benzene and ethylene. The feed consists of a mixture of an ethylene-rich hydrocarbon stream and benzene reactant, under high pressure. The feed is mixed with recycle streams from the process and preheated to the reaction temperature before being fed to the reactor. The reaction temperature is usually in the range 200°F to 300°F. The ALKAR process employs a fixed bed, high pressure catalytic reactor with the reaction pressure in the range of 500 psig. An excess of benzene to ethylene is maintained to favor the selectivity of ethylene to ethylbenzene over higher alkylated derivatives. The formation of ethylbenzene from ethylene and benzene is given by the following exothermic reaction:

$$C_2H_4 + C_6H_6 \rightarrow C_6H_5C_2H_5 \qquad \Delta H = -27.21 \text{ Kcal/mol}$$

The reactor conversion achieved based on ethylene is on the order of 99+%. However, ethylene reacts further with the monoalkylated product to yield a diethylbenzene byproduct according to the reaction:

$$C_2H_4 + C_6H_5C_2H_5 \rightarrow C_2H_5C_6H_4C_2H_5$$

This is clearly undesirable because the higher derivatives need to be transalkylated at a later stage to increase the yield of the ethylbenzene product

The effluent from the reactor is first flashed to remove any light gases from the process. The bottoms of the flash is then fed to the benzene separation column. Excess benzene is separated from the alkylated products and recycled back to the feed from the top of the column. The bottoms of the column, which is primarily a mixture of ethylbenzene and diethylbenzene, is then fed to the ethylbenzene column. This column effects the separation between the mono and dialkylated derivatives.

The Reactor Model

As mentioned above, the ALKAR process uses a fixed-bed catalytic reaction, with a Boron trifluoride (BF_3) catalyst supported on alumina (Al_2O_3). The kinetics of ethylation of benzene have been studied by Govindarao *et a/.* [12]. At high pressure (approx. 500 psig) the reacianls are primarily in the liquid phase and Govindarao *et al.* [12] reported a first order dependence, with respect to each reactant, for formation of both ethylbenzene and diethylbenzene. From their experimental data, the following rate expressions can be derived for the two reactions:

$$r_{EB} = k_1^* \left[C_2H_4\right] \left[C_6H_6\right]$$
$$r_{DI} = k_2^* \left[C_2H_4\right] \left[C_6H_5C_2H_5\right]$$

where

$$k_1^* = 98.3692 \ \exp(-759.3/RT) \quad ftVlbmol\text{-}hr$$
$$k_2^* = 16.021X10^4 \ \exp(-5764.2/RT) \quad ftVlbmol\text{-}hr$$

The mass balance relations for the parallel reactions, based on the conversion of ethylcne. are given by the differential equations corresponding to a plug flow reactor model:

$$F_{E_0}\, dx_{E_1} = K_{EB}\, dV$$

$$F_{E_0}\, dx_{E_2} = A_{DI}\, dV$$

where

F_{E_0} - inlet molar flow of ethylene to reactor (lbmol/ft^3)

x_{E_1} - conversion of ethylene to ethylbenzene

x_{E_2} - conversion of ethylene to diethylbenzene

V - reactor volume

The reactor is assumed to operate isothermally; hence there is no temperature change to affect changes in specific colume. Also energy balances need not be computed in this case. The two differential equations are integrated simultaneously using the Modified Euler method until a prespecified overall conversion for ethylene, $x_E = x_{E_1} + x_{E_2}$, is attained.

Optimization Problem and Results

The objective function was taken as the combined sales of the products, ethylbenzene and diethylbenzene on an annual basis, assuming 8400 operating hours (Table 4). Cost coefficients were obtained by adjusting the product prices based on the data reported for 1971-72 [22].

The constraints imposed on the process are also specified in Table 4. A high benzene to ethylene molar ratio, from 5 to 10, is required to increase the selectivity of ethylene to ethylbenzene. The specification on the upper bound for the total loss of ethylene to diethylbenzene influences this ratio. Also it is desirable to .restrict the loss of benzene from the recycle loop so that the fresh feed of benzene is kept low. The last constraint on the reactor temperature, as noted before, stems from the actual operating conditions of the ALKAR process.

The 6 decision variables and their respective bounds are specified in Table 5. The seven component flows and specific enthalpy of the reactor effluent account for the 8 tear variables. Tear stream pressure need not be included since the loop pressure is determined by the minimum of the feed and outlet pressures from the recycle pumps. The process was optimized using the SPAD simulator on a DEC-20 computer.

The model simplification scheme used in the ethylbenzene problem is similar to the one applied in the flash problems. The SRK equations form the rigorous model with the ideal option acting as its simplified counterpart. For this problem, use of the ideal option as the simplified model represents a very poor approximation to the SRK model. This is because Raoult's Law is applicable to' hydrocarbon systems only at low pressures. However, stream pressures in this problem vary from 6 to 34 atm. Nevertheless, we still use this model because we are primarily interested in observing how well the algorithm performs in reaching the rigorous model solution. As mentioned before, the time saving feature of the algorithm is not considered as significant in judging its performance here because of the poor choice of simple models for this problem.

Results from the rigorous and simplified based optimizations (Table 6) indicate the decision and tear variables as well as the objective function differ, as expected. Table 6 also lists the results from the R/S algorithm. This solution coincides with the optimal objective function from the rigorous model solution, thereby demonstrating the robustness of the algorithm. Here the number of rigorous gradient based steps in this case is 5 compared with the 8 iterations that the complete rigorous model requires (Table 7). Thus, the R/S algorithm clearly exhibits better performance with regard to reducing the rigorous gradient evaluations. It also requires slightly less CPU time (793 sees) in solving the ethylbenzene flowsheet compared to the rigorous model (852 sees). However, this corresponds to only a 7% decrease in the actual CPU time. As seen from Table 7, only the initial simplified model step is acceptable for this problem and all other simplified search directions do not yield descent directions. As discussed for the analytic example above, this performance is not unusual for poor choices of simplified models.

Finally, the restart method performs very poorly for this problem. In fact, it requires even more CPU time than the rigorous model solution itself. Again, this comparison illustrates some advantages of the R/S algorithm over the restart method.

The above results also highlight the need for proper choices of simplified models in order to yield significant time savings. Note that in both flash problems, the flash pressure was restricted to vary over 10 - 50 psia. In this case, the Raoult's Law model automatically acts as a good simplification for the SRK model, because it accurately predicts vapor-liquid equilibrium ratios over this range. Therefore the approximation based algorithm also registers significant savings in CPU time. For the ethylbenzene problem, limitations on the applicability of this model to high pressures make it a poor approximation. This leads to several intermediate points where the simplified model QP solution is not a descent direction for the rigorous model exact penalty. Such directions have to be abandoned and additional expensive rigorous gradients have to be computed. On the other hand, convergence of the algorithm to the rigorous model optimum, in spite of a poor model approximation, clearly demonstrates the robustness of the procedure.

Acknowledgements

Financial support under NSF Grant ENG - 8451058 is gratefully acknowledged for this work.

APPENDIX

Proof of Property 2

The directional derivative for $P(z,\alpha)$ from the solution of the rigorous model QP is given by

$$D_d P(z^i) = \nabla\phi(z^i)^T d + \alpha\Big[\sum_{j\in I} \nabla g_j^T d + \sum_{j\in \bar{I}} (\nabla g_j^T d)_+$$

$$+ \sum_{j\in K} \nabla h_j^T d + \sum_{j\in \bar{K}} |\nabla h_j^T d| - \sum_{j\in \hat{K}} \nabla h_j^T d \Big]$$

where:

$$I = \{ j \mid g_j(z^i) > 0 \}$$

$$\bar{I} = \{ j \mid h_j(z^i) = 0 \}$$

$$K = \{ j \mid h_j(z^i) > 0 \}$$

$$\bar{K} = \{ j \mid h_j(z^i) = 0 \}$$

$$\hat{K} = \{ j \mid h_j(z^i) < 0 \}$$

From the QP solution we have for all j:

$$\nabla h_j^T d = -h_j$$

$$\nabla g_j^T d \le -g_j$$

$$u_j(g_j + \nabla g_j^T d) = 0$$

$$\nabla\phi^T d = -d^T B d - \sum_j u_j \nabla g_j^T d - \sum_j v_j \nabla h_j^T d$$

Substituting these expressions into the directional derivative and using the properties of the index sets yields:

$$D_d P(z^i) \le -d^T B d - \sum_{j\in I} (\alpha - u_j) g_j - \sum_{j\in K\cup\hat{K}} (\alpha - v_j) |h_j|$$

which is negative for $\alpha > \max_{\forall j} (u_j, | v_j |)$.

REFERENCES

[1] Barrett, A. and J. J. Walsh, "Improved Chemical Process Simulation Using Local Thermodynamic Approximations", *Comp. and Chem. Eng.*, 3, 397 (1979)

[2] Biegler, L. T., "Simultaneous Modular Simulation and Optimization", Proc. 2nd Int. Conf. on FOCAPD, 369, CACHE Corp. (1984)

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

[4] Biegler, L. T. and R. R. Hughes, "Infeasible Path Optimization of Sequential Modular Simulators", *AIChE J.*, 28, 994 (1982)

[5] Biegler, L. T., I. E. Grossmann and A. W. Westerberg, "A Note on Approximation Techniques Used in Process Optimization", *Comp. and Chem. Eng.*, 9, 201 (1985)

[6] Boston, J. F. and H. I. Britt, "A Radically Different Formulation and Solution of the Single-Stage Flash Problem", *Comp. and Chem. Eng.*, 2, 109 (1978)

[7] Chamberlain, R. M., M. J. D. Powell, C. Lemarechal and H. C. Pedersen, "The Watchdog Method for Forcing Convergence in Algorithms for Constrained Optimization", *Math. Prog. Study*, 16, 1 (1982)

[8] Chimowitz, E. H., Anderson, T. F., Macchietto, S. and L. F. Stutzman, "Local Models for Representing Phase Equilibria in Multicomponent Nonideal Vapor-Liquid and Liquid-Liquid Systems. 1. Thermodynamic Approximation Functions", *I & EC Proc. Des. & Devt.*, 22, 217 (1983).

[9] Chimowitz, E. H., Anderson, T. F., Macchietto, S. and L. F. Stutzman, "Local Models for Representing Phase Equilibria in Multicomponent Nonideal Vapor-Liquid and Liquid-Liquid Systems. 2. Application to Process Design", *I & EC Proc. Des. & Devt.*, 23, 609 (1984).

[10] Coleman, T. F. and A. R. Conn, "Nonlinear Programming Via an Exact Penalty Function: Global Analysis", *Math. Programming.* 24, 137 (1982)

[11] Evans, L. B., R. A. Trevino-Lozano, H. I. Britt and J. F. Boston, "Simultaneous Modular Process Simulation and Optimization", IChemE Symp. Ser. #92, 25 (1985)

[12] Govindarao, V. M. H., P. K. Deshpande and N. R. Kuloor, "Kinetics of Ethylation of Benzene", *Ind. J. Tech.*, 4, 234 (1966)

[13] Grens II, E. A., "Efficient Use of Thermodynamic Models in Process Calculations", Proc. 2nd Int Conf. on **FOCAPD,** 249, CACHE Corp. (1984)

[14] Han, S-P, "A Globally Convergence Method for Nonlinear Programming", *J. Opt. Theory and Appl.*, 22, 297 (1977)

[15] Han, S-P and O. L. Mangasarian, "Exact Penalty Functions in Nonlinear Programming", *Math. Programming,* 17, 251 (1979)

[16] Hughes, R. R., R. K. Malik and L. T. Biegler, "Simulator for Process Analysis and Design", EES Report # 52, University of Wisconsin, Madison (1981)

[17] Jirapongphan, S., ScD Thesis, MIT, Cambridge, MA (1980)

[18] 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).

[19] Leesley, M. E. and G. Heyen, "The Dynamic Approximation Method of Handling Vapor-Liquid Equilibrium Data In Computer Calculations for Chemical Processes", *Comp. and Chem. Eng.*, 1, 109 (1977)

[20] McKetta, J. J. (ed.), *Encyclopedia of Chemical Processing and Design,* 20, 77, M. Dekker, New York (1984)

[21] Maratos, N., Ph. D Thesis, University of London (1978).

[22] Noble, P. (ed.), *The Kline Guide to Chemical Industry*, C Kline & Co., Fairfield. NJ (1974)

[23] Powell, M. J. D., "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations", presented at 1977 Dundee Conf. on Num. Anal. (1977)

[24] Schittkowski, K., "The Nonlinear Programming Algorithm of Wilson, Han and Powell with an Augmented Lagrangian Type Line Search, Part I : Convergence Analysis", *Numer. Math*, 38, 83 (1982).

[253 **Trevino-Lozano, R., Ph. D** Thesis, DepL of Chemical **Eng., MIT, Cambridge,** MA (1985)

Table 1

Flash Recycle Flowsheet Optimization : Problem Definition

FEED DATA

COMPONENT	FEED FLOWRATE (mols/hr)
Propane	10.0
1-Butene	15.0
N-Butane	20.0
Trans-2-Butene	20.0
Cis-2-Butene	15.0
N-Pentane	10.0

Feed Press : 1034.25 KPa

Feed Temp : 37.78 C

Decision Variables : **Splitter Ratio**, x_1 $(0.2 \le x_1 \le 0.8)$

Flash Pressure, x_2 $(69 \le x_2 \le 345)$ KPa

Tear Variables : **Recycle flows**, y_i

$(0 \le y_i \le 100)$ mols/hr

Specific Enthalpy of Recycle, H

$(-1.055E+07 \le H \le 1.055E+07)$ J/mol

OBJECTIVE FUNCTION

Monotonic : $\max\ e_1$

Nonlinear : $\max\ e_1^2 e_2 - e_1^2 - e_3^3 + e_4 - e_5^{0.5}$

(e_i - component **flow** in Flash Overhead)

Table 2

Performance of R/S Algorithm

MONOTONIC OBJECTIVE FLASH PROBLEM

Rigorous Model : Soave-Redlich-Kwong

Simplified Model : Raoult's Law, Antoine's Equation, Ideal Enthalpy

PROCEDURE	NO. OF ITERS	NO.-RIG STEPS	NO.-SIM STEPS	OBJEC FUNC	DECSN VAR	CPU SEC	KKT TOL
COMPLETE RIGOROUS MODEL	8	8	-	7.32252	(0.2,10)	184	1E-03
COMPLETE SIMPLIFIED MODEL	5	-	5	7.28787	(0.2,10)	1 0	1E-03
COMBINED RIG-SIMP. (R/S) MODEL	12*	5	2	7.32240	(0.2,10)	135	1E-03
RESTART METHOD or (TWO-STAGE METHOD)	9	4	5	7.32240	(0.2,10)	79	1E-03

Starting Point : (0.5,40)

* - No. of Iterations = 2 (No. of Rig. Step
+ No. of Simp. Steps

Table 3

Performance of R/S Algorithm

NONLINEAR OBJECTIVE FLASH PROBLEM

Rigorous Model : Soave-Redlich-Kwong

Simplified Model : Raoulfs Law, Antoine's Equation, Ideal Enthalpy

PROCEDURE	NO. OF ITERS	NO.-RIG STEPS	NO.-SIM STEPS	OBJEC FUNC	DECSN VAR	CPU SEC	KKT TOL
COMPLETE RIGOROUS MODEL	11	11	-	3.66289	(0.8) (18.96)	232	10^{-3}
COMPLETE SIMPLIFIED MODEL	13	-	13	4.87575	(0.8) (23.79)	29	10^{-3}
R/S ALGORITHM	15*	7	1	3.66308	(0.8) (18.99)	184	10^{-3}
RESTART METHOD	21	8	13	3.66337	(0.7999) (19.02)	193	10^{-3}

Starting Point : (0.5, 40)

* - No. of Iterations = 2 (No. of Rig. Steps)
+ No. of Simp. Steps

Table 4

Ethylbenzene Flowsheet

OBJECTIVE FUNCTION AND CONSTRAINTS

Ojective function

$$O = 8400 * (C_{EB} * F_{EB} + C_{DIEB} * F_{DIEB})$$

where

C_{EB} =$ 7.42/lbmol; C_{DIEB} = $ 5.36/lbmol

Constraints

1. Benzene / Ethylene in Reactor Feed, R : 5 <$ R ^ 10

2. Loss of Ethylene to DIEB ^ 10 % of Ethylene Converted

3. Loss of Benzene from Process ^ 2 % of Benzene in Reactor feed

4. Reactor inlet Temperature, T : 200 >< T ^ 250 (DEC F)

EB = Ethylbenzene, DIEB = Diethylbenzene

F = Flowrate (mol/hr) from bottom of Benzene column

C = Cost coefficient for the component

Table 5

ETHYLBENZENE FLOWSHEET: PROBLEM VARIABLES

DECISION VAR.	BOUNDS	START PT.
BLEED SPL. FRAC TO RECYCLE	$0.9 < X_1 < 0.99$	0.99
BENZENE COLUMN PRESS (PSIA)	$60 < X_2 < 95$	60.
LIGHT KEY FRAC TO TOPS (BENZENE COL)	$0.95 < X_3 < 0.995$	0.995
HEAVY KEY FRAC TO BTMS (BENZENE COL)	$0.95 < X_4 < 0.995$	0.995
HEAVY SPLIT. FRAC TO BENZ COL.	$0.5 < X_5 < 0.9$	0.9
FLASH PRESS (PSIA)	$20 < X_6 < 95$	75.0

TEAR VARIABLES : FLOWRATES AND SPECIFIC ENTHALPY IN INLET STREAM TO FLASH (NO. = 8)

STARTING PT : (0.25, 43, 43, 78, 430, 46, 5)

FEED DATA :

COMPONENTS	FLOWRATE (lb mol/hr)
ETHYLENE	50
PROPYLENE	5
1-BUTENE	2
1-PENTENE	2
BENZENE	55

FEED PRESSURE : 500 PSIA

FEED TEMP : 100 $^{\circ}$F

Table 6

Performance of new Algorithm

Ethylbenzene Problem - Optimal Solutions

	RIGOROUS METHOD	SIMPLIFIED METHOD	R/S ALGORITHM	RESTART METHOD
OBJEC. FN. (S/YR)	2.77242X10*	$2.77102X10^{6}$	$2.77242X10^{6}$	$2.77242X10^{6}$
DECISION VARIABLES				
1. BLEED FRAC	0.9855	0.9861	0.9855	0.9855
2. BENZ. COL PRESS (PSIA)	60.007	60.018	60.146	60.000
3. LIGHT KEY FRACTN	0.995	0.9937	0.995	0.995
4. HEAVY KEY FRACTN	0.995	0.995	0.995	0.995
5. HEAVY FRAC TO BZ COL.	0.900	0.900	0.900	0.900
6. FLASH PRESS	95.00	95.00	95.00	95.00
TEAR VARIABLES				
1. ETHYLENE	0.504	0.504	0.504	0.504
2. PROPYLENE	65.419	65.339	65.419	65.418
3. 1-BUTENE	49.589	59.811	49.573	49.573
4. 1-PENTENE	83.017	92.170	83.017	83.017
5. BENZENE	458.443	458.311	458.441	458.444
6. ETHYLBZ	46.263	46.250	46.263	46.263
7. DIETHYLBZ	4.710	4.706	4.710	4.710
SP. ENTHALPY	-2166.667	7435.160	-2166.667	-2166.667

Table 7

Performance of R/S Algorithm

ETHYLBENZENE FLOWSHEET - PERFORMANCE

Rigorous Model : Soave-Redlich-Kwong

Simplified Model : Raoult's Law, Antoine's Eqn, Ideal Enthalpy

PROCEDURE	NO.-RIG STEPS	NO.-SIM STEPS	CPU SEC	KKT TOL
COMPLETE RIGOROUS MODEL	8	-	852	10^{-3}
COMPLETE SIMPLIFIED MODEL	-	7	133	10^{-3}
COMBINED RIG-SIMP.	5	1	793	10^{-3}
RESTART METHOD	9	7	1098	10^{-3}

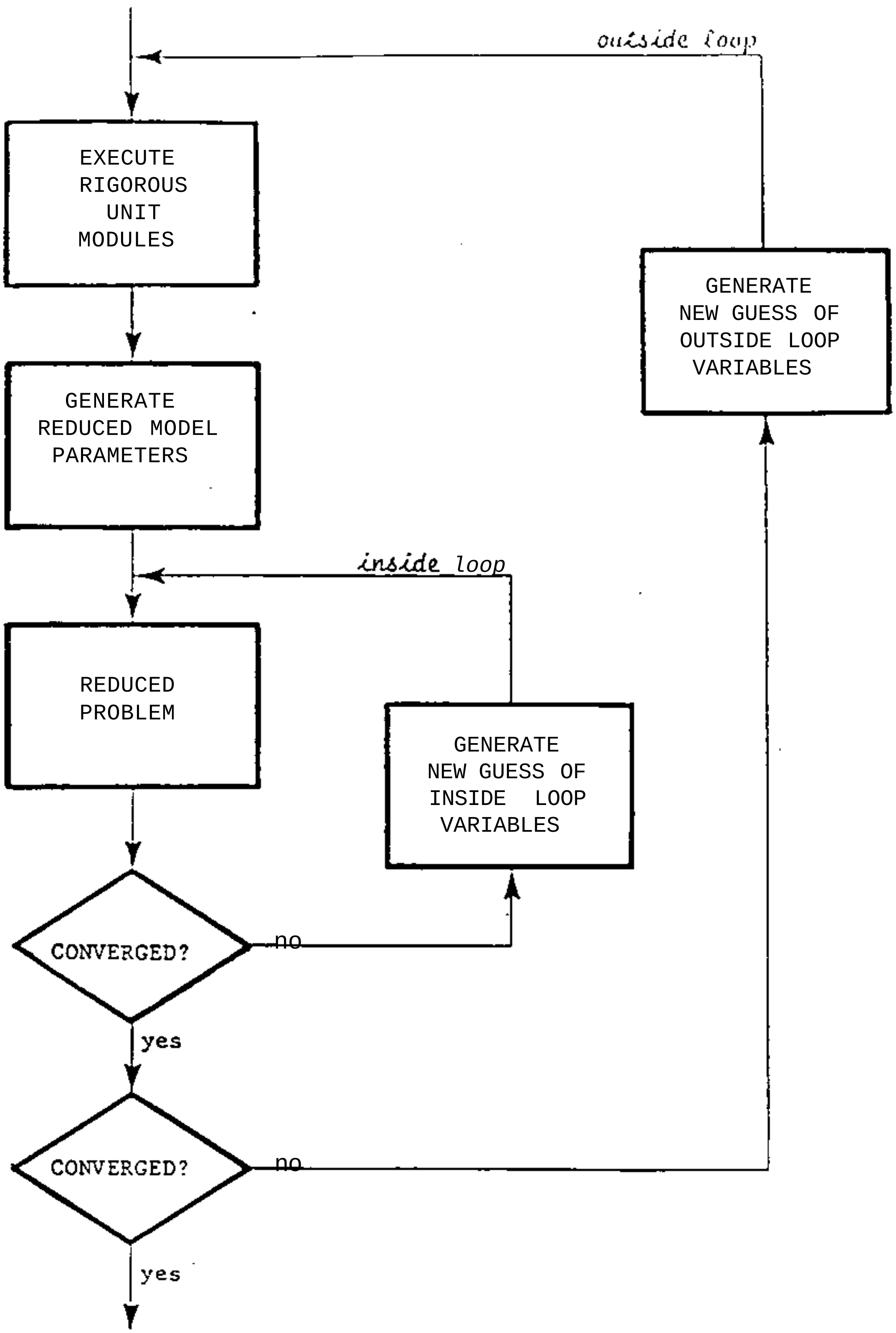
FIGURE CAPTIONS

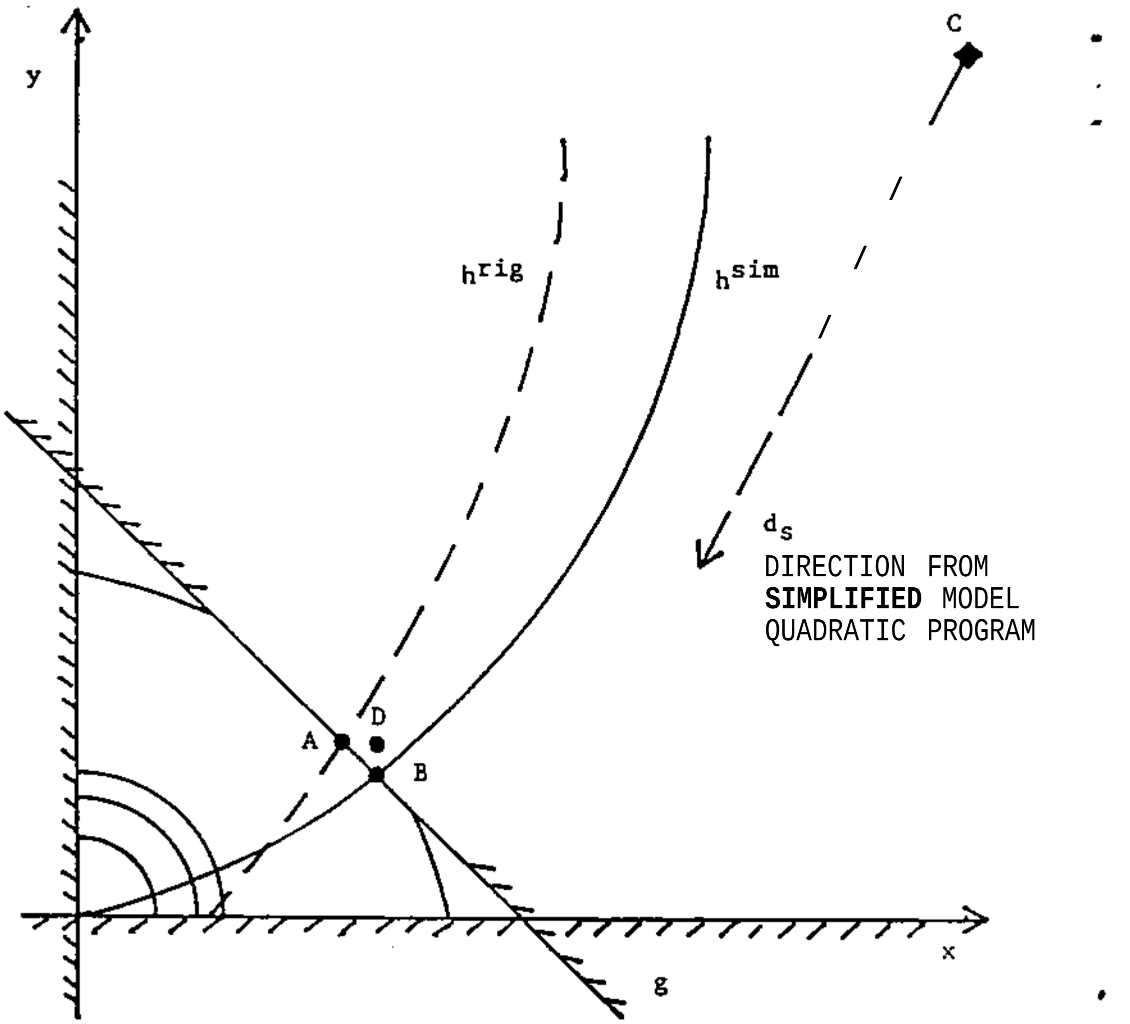
Figure 1: Calculation Sequence for "Inside-Out" Optimization Algorithm

Figure 2: Analytic Example Problem for R/S Algorithm

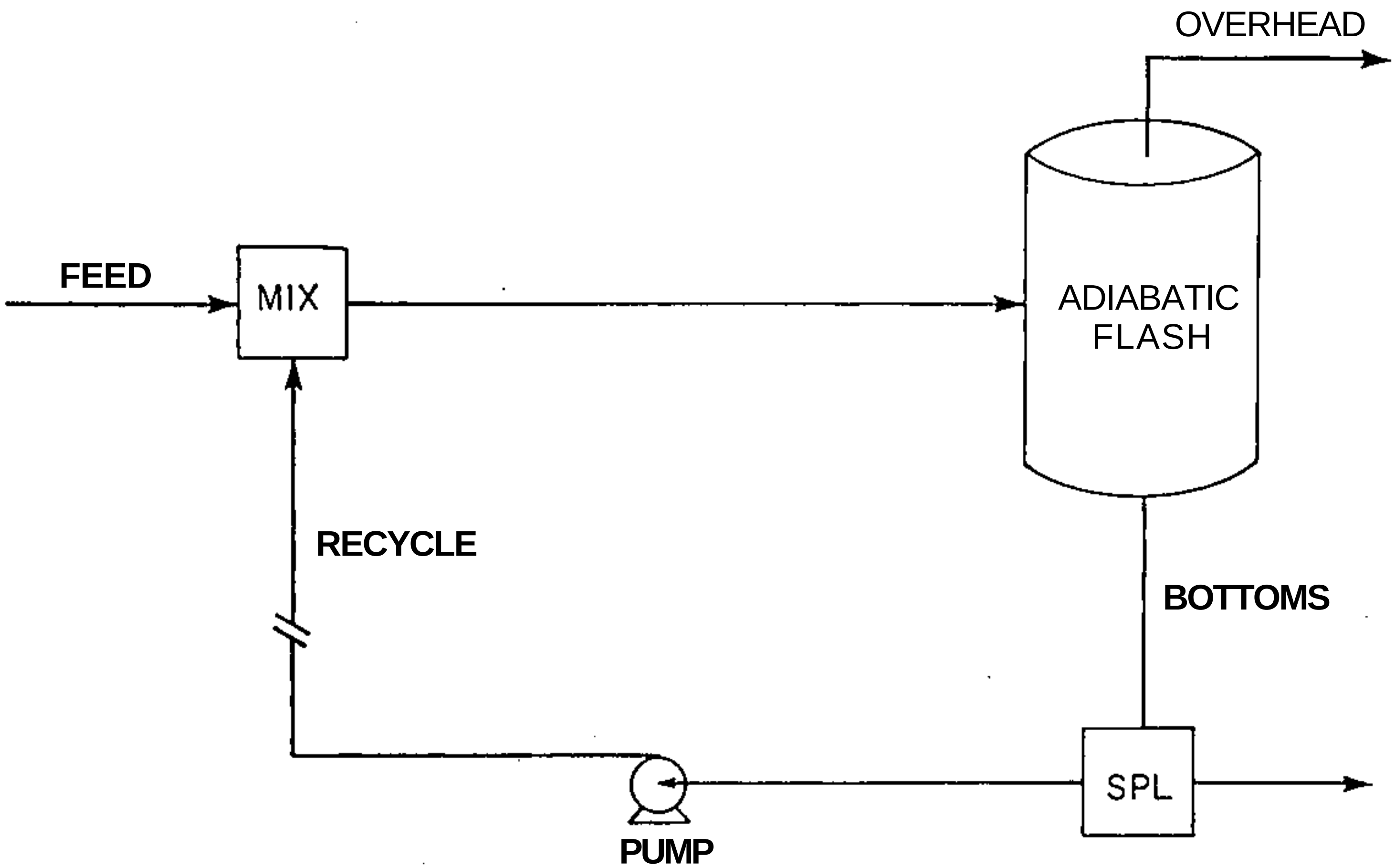
Figure 3: Simple Flash Recycle Flowsheet

Figure 4: Ethylbenzene Process Flowsheet

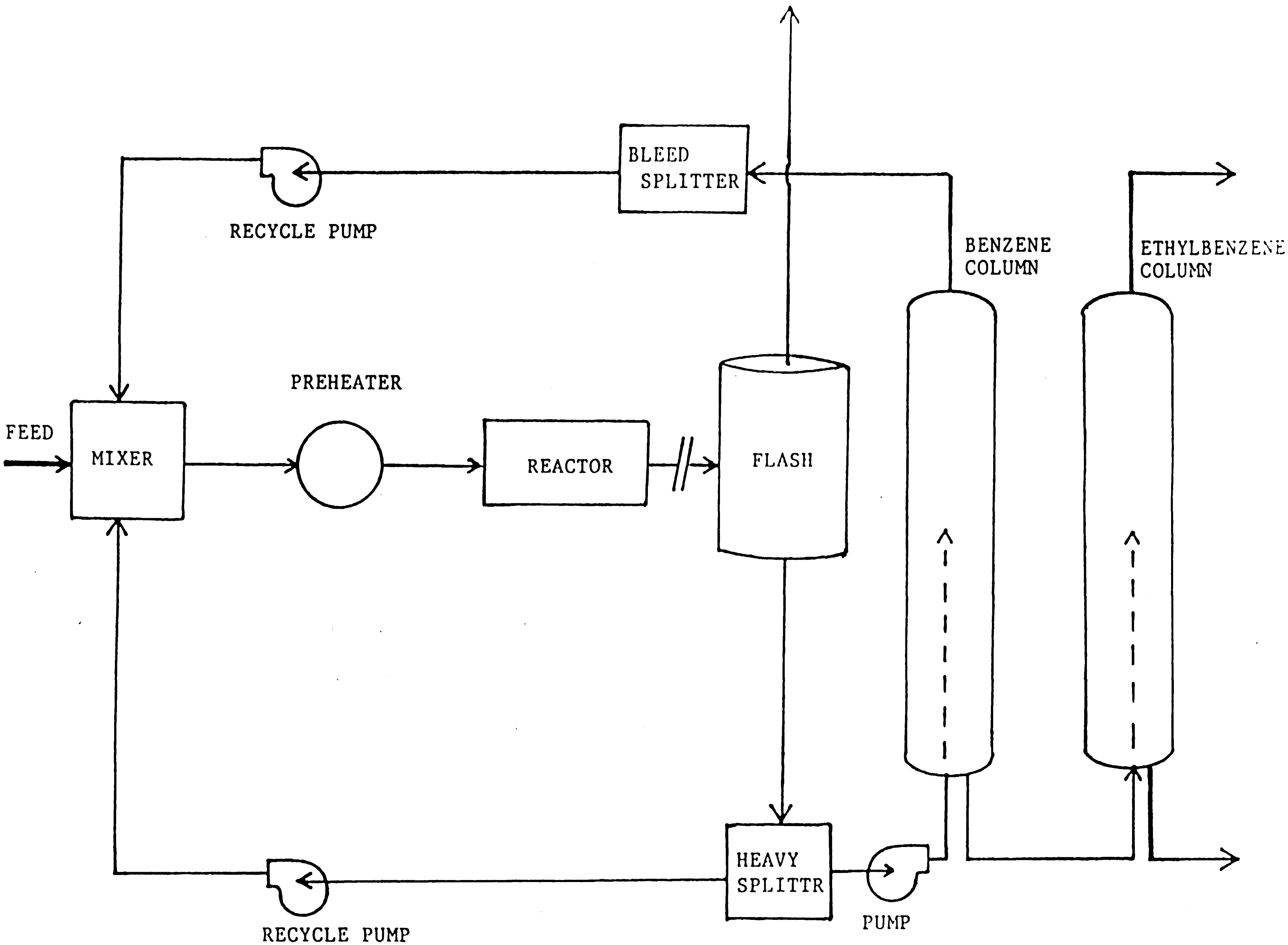

outside loop
EXECUTE RIGOROUS UNIT MODULES
GENERATE REDUCED MODEL PARAMETERS
inside loop
REDUCED PROBLEM
GENERATE NEW GUESS OF INSIDE LOOP VARIABLES
GENERATE NEW GUESS OF OUTSIDE LOOP VARIABLES
CONVERGED?
no
yes
CONVERGED?
no
yes



A - RIGOROUS MODEL OPTIMUM
B - SIMPLIFIED MODEL OPTIMUM



SIMPLE FLASH RECYCLE FLOWSHEET


BLEED
SPLITTER
RECYCLE PUMP
BENZENE
COLUMN
ETHYLBENZENE
COLUMN
PREHEATER
FEED
MIXER
REACTOR
FLASH
HEAVY
SPLITTR
PUMP
RECYCLE PUMP