# Efficient Handling of the Implicit Constraints Problem for the ASPEN MINLP Synthesizer

**U. M. Diwekar* and E. S. Rubin**

*Environmental Institute & Department of Engineering and Public Policy, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213*

A new process synthesizer has been recently implemented in the ASPEN chemical process simulator. The process synthesizer determines optimal flowsheet configurations and is based on a mathematical programming (MINLP) optimization algorithm. This MINLP algorithm consists of solving alternating sequences of NLP subproblems and MILP master problems. The NLP subproblem provides the linearization information of the nonlinear constraints which relate the output variables to input variables specified explicitly. For a simulator like ASPEN, most of the relations are implicit, which prevents the NLP optimizer from transferring this crucial information to the master problem. In our earlier work, a strategy was outlined to circumvent this problem by adding additional variables and constraints to the NLP problem. This procedure invariably increases the load on the NLP optimizer, which is normally the efficiency-determining factor in the large-scale MINLP process synthesis. In this paper, we present a new and efficient strategy for handling these additional variables and constraints by means of partitioning the variables. This approach is shown to decrease the computational time significantly.

## 1. Introduction

The mathematical programming approach to process synthesis involves (a) formulation of a flowsheet superstructure incorporating all the alternative process configurations and (b) modeling the superstructure as an MINLP problem of the form

MINLP:

$$Z = \min_{\bar{x},\bar{v},\bar{y}} c^{\mathrm{T}}\bar{y} + f(\bar{x},\bar{v})$$

subject to

$$h(\bar{x},\bar{v}) = 0$$

$$h1(\bar{x},\bar{v}) = v - z(\bar{x}) = 0$$

$$B^{\mathrm{T}}\bar{y} + g(\bar{x},\bar{v}) \leq 0$$

$$y \in Y; \quad x \in X$$

where

$$Y = [y|Ay \leq a, y[0,1]^m]; \quad X = [x|x^{\mathrm{L}} \leq x \; Le \; x^{\mathrm{U}}]$$

The continuous variables $x$ represent flows, operating conditions, and design variables. The variables $v$ are the output variables which are related to the input variables $x$ by model equations. For equation-oriented environments, these model equations are embedded in the equality constraints $h1(\bar{x},\bar{v})$. The binary varibles $y$ denote the potential existence of process units. The mathematical programming approach also involves (c) identification of both the optimal configuration and operating process parameters by an algorithm based on an alternating sequence of Nonlinear Programs (NLPs) and Mixed Integer Linear Programs (MILPs).

Over the past few years, significant advances in mixed integer nonlinear programming (MINLP) algorithms have led to rapid developments of equation-oriented software packages such as APROS (Paules and Floudas, 1989), DICOPT++ (Kocis and Grossmann, 1989a; Viswanathan and Grossmann, 1990), and PROSYN (Kravanja and Grossmann, 1990). These packages are based either on variants of Generalized Benders Decomposition (Floudas et al., 1989) or on variants of Outer Approximation (Kocis and Grossmann, 1987; Viswanathan and Grossmann, 1990), where the zero flow problems encountered sometimes in the process synthesis can be handled using the decomposition strategy proposed by Kocis and Grossmann (1989b). Alternatively, one can use the recently proposed MSGA algorithm (Salcedo, 1992) which does not need the decomposition strategy and is robust in the face of nonconvexities. Although these packages provide an environment for solving MINLP process synthesis problems, they have some practical limitations; for example, it is difficult to solve process synthesis problems involving complex chemical processes. Sequential modular simulators like ASPEN, PROCESS, PROII, etc., are more widely used in chemical industries than equation-oriented simulators. Such simulators have grown in sophistication over the years and have useful capabilities for modeling many complex chemical processes. Therefore, it is more desirable to build the MINLP process synthesis capability around such simulators.

A new process synthesis capability built around the public version of the ASPEN simulator (Diwekar et al., 1991) represents a step in this regard. The implementation of this new capability in a sequential modular simulator poses challenging problems which are not encountered in equation-oriented simulators; therefore, new strategies are necessary to solve these problems. One such problem associated with the MINLP sequential modular process synthesizer, which this paper addresses, is that of *implicit constraints.*

The problem of implicit constraints is encountered in sequential modular simulators because of the black box nature of the models in sequential modulator simulators. The ASPEN MINLP environment is based on a two-level optimization algorithm consisting of an upper-level MILP master problem and a lower-level NLP problem, as shown in Figure 1. It is realized that the problem of implicit constraints holds for all MINLP solvers, in general, and is not restricted to MINLP solvers that depend on an alternate sequence of NLPs and MILPs. However, since

0888-5885/93/2632-2006$04.00/0 

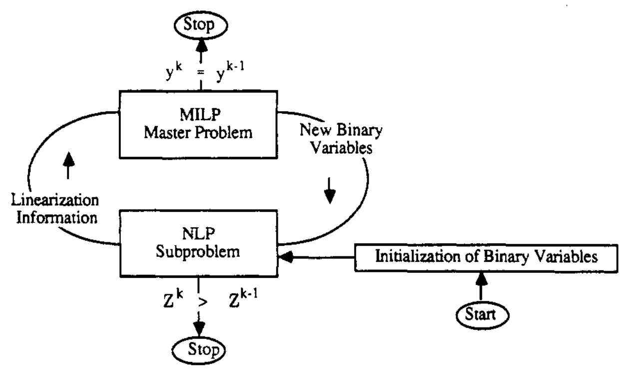



**Figure 1.** Main steps in the MINLP algorithm.

the paper focuses on the ASPEN MINLP synthesizer which is based on the MINLP solvers that depend on an alternate sequence of NLPs and MILPs, the implicit constraint problem is explained in terms of this algorithm. The MILP master problem predicts new binary variables, while the NLP problem provides new continuous variables. The MILP master problem represents the linearized NLP problem with nonfixed binary varibles, since at each stage the MILP master problem obtains the linearization information from the NLP optimizer. In sequential modular simulators, most of the nonlinear constraints are not represented explicitly by equations, unlike equation-oriented simulators. The linearization information on these constraints, which are essentially black box relations embedded in the simulator environment, therefore must be passed to the master problem.

In order to circumvent this problem of implicit constraints, new decision varibles are created, and these are equated to the output variables from the flowsheet configurations. This procedure ensures that the original MINLP problem remains the same, while at each stage the MILP master problem receives increased information from the NLP optimizer. As can be expected, although this procedure assures complete information transfer to the master problem, it also increases the computational load on the NLP optimizer, which is generally the rate-determining step in the MINLP process synthesis. In this paper, we exploit the natural partitioning of the real decision variables and pseudo decision variables to solve the NLP subproblem. This strategy reduces the computational load on the NLP problem crucial for the solution of large-scale synthesis problems.

## 2. Implicit Constraint Problem

In an equation-oriented environment like PROSYN, all the nonlinear equality constraints are specified explicitly. While, for a sequential modular simulator (SMS) like ASPEN, most of the constraints are implicit as shown below (SMS MINLP). This includes the black box relation between the output variables $v$ (which are part of the objective function or constraints in the master problem) and the input decision variables $x$. During the solution cycle, these implicit relations are not transferred to the master problem, resulting in a suboptimal solution.

SMS MINLP:

$$Z = \min_{\bar{x},\bar{y}} c^{\mathrm{T}}\bar{y} + f(\bar{x},\bar{v})$$

subject to

$$h(\bar{x},\bar{v}) = 0$$

$$B^{\mathrm{T}}\bar{y} + g(\bar{x},\bar{v}) \leq 0$$

$$y \in Y; \quad x \in X$$

The following example illustrates this implicit constraint problem that is encountered while solving the MINLP process synthesis problem using a sequential modular simulator.

**Example.** Figure 2a shows the two-reactor problem considered by Kocis and Grossmann (1989b) which in its original form is represented by the following equations:

$$\text{minimize } Z = 7.5y_1 + 5.5y_2 + 7v_1 + 6v_2 + 5x$$

subject to

$$z_1 = 0.9[1 - \exp(-0.5v_1)]x_1$$

$$z_2 = 0.8[1 - \exp(-0.4v_2)]x_2$$

$$x_1 + x_2 - x = 0$$

$$z_1 + z_2 = 10$$

$$v_1 \leq 10y_1$$

$$v_2 \leq 10y_2$$

$$x_1 \leq 20y_1$$

$$x_2 \leq 20y_2$$

$$y_1 + y_2 = 1$$

$$y_1, y_2 = 0, 1 \quad \text{and} \quad x_1, x_2, z_1, z_2, v_1, v_2 \geq 0$$

where $y_1$ and $y_2$ are binary variables representing the presence and absence of reactors 1 and 2, respectively, and $z_1$ and $z_2$ represent the outputs from the reactors as a function of reactor volumes $v_1$ and $v_2$ and inlet flows $x_1$ and $x_2$.

Figure 2b shows the sequential modular (ASPEN) representation of the same problem with the MINLP process synthesizer. It may be noted that, in this formulation, the output variables $z_1$ and $z_2$ are correlated to the inputs by black box relations (Figure 2b) which are not transparent to the NLP optimizer. Hence, in contrast to the formulation above, explicit relations for $z_1$ and $z_2$ do not appear in the MINLP form given below:

$$\text{minimize } Z = 7.5y_1 + 5.5y_2 + 7v_1 + 6v_2 + 5x$$

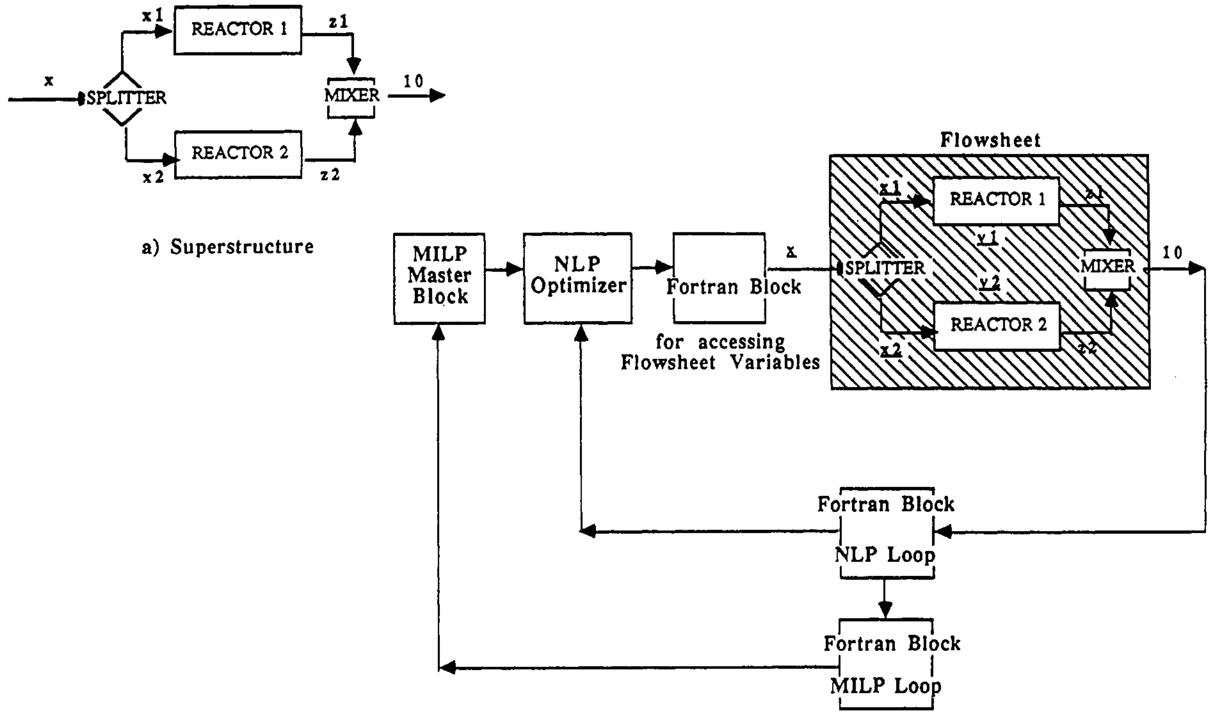
subject to

$$x_1 + x_2 - x = 0$$

$$v_1 \leq 10y_1$$

$$v_2 \leq 10y_2$$

$$x_1 \leq 20y_1$$




a) Superstructure

b) ASPEN Representation

**Figure 2.** The two-reactor problem.

$$x_2 \le 20y_2$$

$$y_1 + y_2 = 1$$

$$y_1, y_2 = 0,1 \quad \text{and} \quad x_1, x_2, v_1, v_2 \ge 0$$

As stated earlier, the MINLP process synthesis involves solving alternating sequences of NLP subproblems and MILP master problems (Figure 1). Accordingly, the new formulation leads to the following steps:

*I. Initial NLP Formulation.* The NLP optimization of the initial flowsheet with $(y_1,y_2) = (0,1)$ is given by

$$\text{minimize } Z = 5.5 + 6v_2 + 5x$$

subject to

$$x_2 - x = 0$$

$$v_2 \le 10$$

$$x_2 \le 20$$

The solution to this NLP is $Z = 107.376$ at $x_2 = 15$ and $v_2 = 4.479$. In the original formulation, some of the nonlinear constraints involving the variables $z_1$ and $z_2$ needed the decomposition strategy by Kocis and Grossmann (1989b) to transfer the information on nonexisting or disappearing units. By way of contrast, in this formulation none of the nonlinear constraints is associated solely with the disappearing units, and hence the decomposition strategy is not required. The linear approximation of the MINLP at the above NLP subproblem solution results in the following MILP master problem:

*II. MLP Master Problem.*

$$\text{minimize } Z = 7.5y_1 + 5.5y_2 + 7v_1 + 6v_2 + 5x$$

subject to

$$x_1 + x_2 - x = 0$$

$$v_1 \le 10y_1$$

$$v_2 \le 10y_2$$

$$x_1 \le 20y_1$$

$$x_2 \le 20y_2$$

$$y_1 + y_2 = 1$$

$$y_1, y_2 = 0,1 \quad \text{and} \quad x_1, x_2, v_1, v_2 \ge 0$$

It can also be noted that since the MINLP from the ASPEN formulation is linear for this problem, the MILP master problem is identical to the original MINLP. The MILP master problem given above resulted in the suboptimal solution of $y_1 = 0, y_2 = 1$, and hence the algorithm failed to find the optimal solution to the two-reactor problem.

The above example problem demonstrated an important point about MINLP process synthesis using a sequential modular simulator: the implicit (black box) nature of the simulator results in the NLP optimizer being unable to transfer crucial linearization information to the master problem. As a consequence, the algorithm fails to find the optimal solution. A viable alternative to circumvent this problem is to make the implicit constraints explicit, as discussed below.

## 3. Constraints Made Explicit

The implicit constraints can be made explicit by using additional state variables and constraints at the NLP optimization level. This involves equating the vector of pseudovariables $x'$ to the vector of output variables $v$ obtained from the flowsheet to make the black box relations transparent, resulting in the following MINLP with constraints made explicit:

$$Z = \min_{\bar{x},\bar{x}',\bar{y}} c^{\mathrm{T}}\bar{y} + f(\bar{x},\overline{x'})$$

subject to

$$h(\bar{x},\overline{x'}) = 0$$

$$h'(x,\overline{x'}) = \overline{x'} - v = 0$$

$$B^{\mathrm{T}}\bar{y} + g(\bar{x},\overline{x'}) \leq 0$$

$$y \in Y; \quad x \in X$$

The procedure is explained below using the two-reactor problem. We first add to the original decision vector, $\bar{x}$, pseudovariables $x'_1$ and $x'_2$ to get an augmented vector:

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \\ x'_1 \\ x'_2 \end{bmatrix}$$

Similarly, we then modify the equality constraint vector $h(\bar{x},\bar{v})$ to incorporate the pseudovariables $x'_1$ and $x'_2$:

$$H(\bar{x}) = [x_1 + x_2 - x = 0] \rightarrow \begin{bmatrix} x_1 + x_2 - x = 0 \\ x'_1 + x'_2 = 10 \\ x'_1 - z_1(x_1,v_1) = 0 \\ x'_2 - z_2(x_2,v_2) = 0 \end{bmatrix}$$

Using this transformation, the two-reactor problem may now be written as follows.

$$\text{minimize } Z = 7.5y_1 + 5.5y_2 + 7v_1 + 6v_2 + 5x$$

subject to

$$x_1 + x_2 - x = 0$$

$$x'_1 - z_1(x_1,v_1) = 0$$

$$x'_2 - z_2(x_2,v_2) = 0$$

$$x'_1 + x'_2 = 10$$

$$v_1 \leq 10y_1$$

$$v_2 \leq 10y_2$$

$$x_1 \leq 20y_1$$

$$x_2 \leq 20y_2$$

$$y_1 + y_2 = 1$$

$$y_1,y_2 = 0,1 \quad \text{and } x_1, x_2, x'_1, x'_2, v_1, v_2 \geq 0$$

It can be seen that this new formulation is equivalent to the original equation-oriented form of the two-reactor problem. The MINLP process synthesizer provides the optimal solution, $y_1 = 1$, $y_2 = 0$, and $Z = 99.240$, in one MILP and two NLP iterations.

## 4. Variables and Constraints Partitioning

In the example above, a strategy to handle the problem of implicit constraints encountered in a sequential modular environment was outlined. Since this strategy includes additional variables and constraints to be incorporated, especially at the NLP stage, it increases the computational time considerably. Most of the NLP optimization techniques commonly used are based on a quasi-Newton method, the robustness and efficiency of which depend on derivative evaluations (Gill et al., 1981). Presently, the sequential modular simulators rely on the method of direct loop perturbations for calculating the derivatives. Each decision variable is perturbed, and the full simulation is carried out to obtain the partial derivatives of the objective function and constraints. This is the most time-consuming part of the NLP optimization. The increased number of variables implies an increased number of perturbations and simulations of the complete flowsheet. To reduce the computational effort associated with the additional variables and constraints, a partitioning strategy is proposed. The decision variable vector $\bar{x}$ is first divided into two parts:

$$\bar{x} = \left[\frac{x}{x'}\right]$$

where the elements $x$ are the real decision variables and $x'$ are the pseudo decision variables which are added to circumvent the implicit constraint problem. As seen earlier in section 3, addition of these pseudovariables calls for additional constraints. The augmented equality constraint matrix ($\bar{x}$) may also be partitioned as follows:

$$H(\bar{x}) = \left[\frac{h(\bar{x},\overline{x'})}{h'(\overline{x'},\bar{x}) = \overline{x'} - v}\right]$$

Since the vector of pseudovariables $x'$ is equated to the vector of output variables obtained from the flowsheet to make the black box relation transparent, they can be considered independent of the flowsheet. In other words, their relation to the flowsheet is through the equality constraints $h'(\overline{x'},\bar{x})$. Therefore, perturbation of these quantities will not have any effect on the flowsheet simulation, and the derivatives of the objective function and the constraints with respect to the additional problem can be provided directly. The Jacobian matrices for the objective function and the constraints may be written as

$$\frac{\partial Z(\bar{x})}{\partial \bar{x}} = \left[\frac{\left(\frac{\partial Z}{\partial \bar{x}}\right)_{\text{perturbation}}}{\left(\frac{\partial Z}{\partial x'}\right)_{\text{analytical}}}\right]$$

$$\frac{\partial H(\bar{x})}{\partial \bar{x}} = \left[\frac{\left(\frac{\partial h}{\partial x}\right)_{\text{perturbation}} \quad \left(\frac{\partial h}{\partial x'}\right)_{\text{analytical}}}{\left(\frac{\partial h'}{\partial \bar{x}}\right)_{\text{perturbation}} \quad I}\right]$$

$$\frac{\partial g(\bar{x})}{\partial \bar{x}} = \left[\frac{\left(\frac{\partial g}{\partial \bar{x}}\right)_{\text{perturbation}}}{\left(\frac{\partial g}{\partial x'}\right)_{\text{analytical}}}\right]$$

This strategy reduces the computational effort considerably. For example, the simple two-reactor problem given in the previous sections required 20% less CPU time on a Vax-3200 when the partitioning strategy was used. The power of this approach is further illustrated in the context

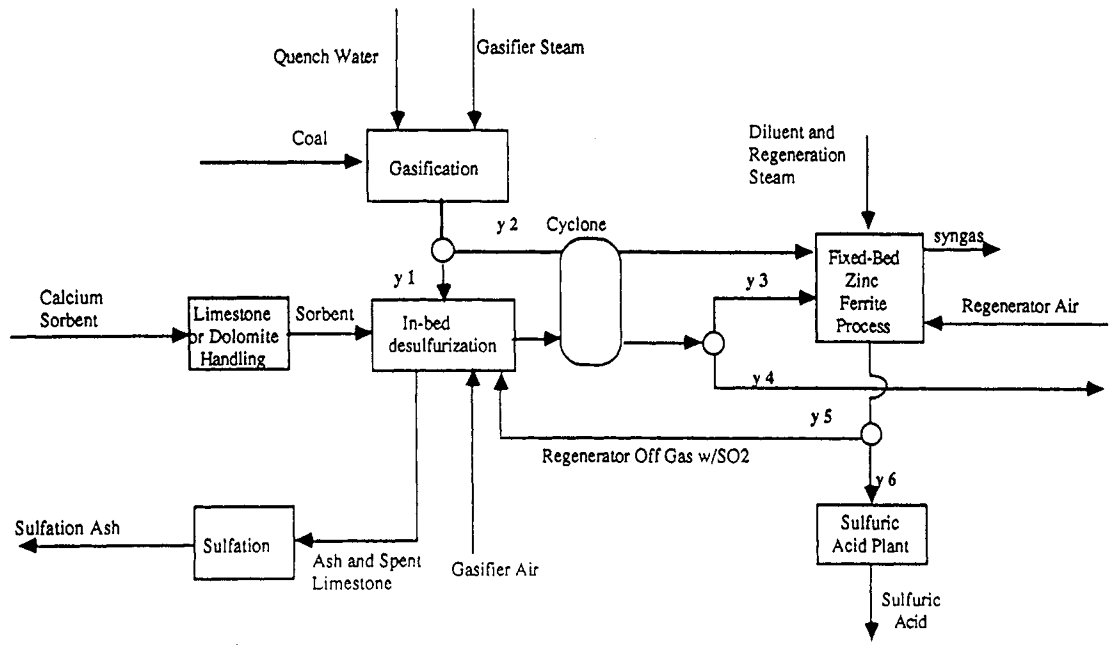



**Figure 3.** Superstructure for the desulfurization section in the KRW IGCC system.

of a large-scale environmental control process synthesis problem elaborated below.

## 5. Application to Environmental Control Design

In recent work by the authors (Diwekar et al., 1992), the new process synthesis capability was used to select an optimal plant configuration and optimal design parameters for an Integrated Gasification Combined Cycle (IGCC) electric power generation system using hot gas cleanup. The superstructure for the desulfurization section is shown in Figure 3, which contains three alternative configurations for sulfur removal, i.e., in-bed desulfurization without recycle ($y_1 = 1$, $y_4 = 1$), gas stream desulfurization ($y_2 = 1$, $y_6 = 1$), and a hybrid system combining both methods ($y_1 = 1$, $y_3 = 1$, $y_5 = 1$). The details of this superstructure are omitted here for the sake of brevity, and the interested reader is referred to Diwekar et al. (1992).

The MINLP formulation of the superstructure to minimize overall cost results in the following equations:

$$\min_{x_1,x_2,x_3} Z = C_{\text{elec}}$$

subject to (a) the emission limit constraint

$$E_{\text{SO}_2} \leq \frac{0.015 \text{ lb SO}_2}{10^6 \text{ Btu}}$$

(b) the logical constraints which are related to the binary variables (Figure 3)

$$y_1 + y_2 = 1$$

$$y_3 + y_4 = y_1$$

$$y_5 + y_6 = 1 - y_4$$

$$y_6 = y_2$$

and (c) the bounds on the decision varibles $x_1$, $x_2$, and $x_3$.

The various terms in the above equations are explained as follows:

$C_{\text{elec}} = \dfrac{C_{\text{T-H}_2\text{SO}_4} + C_{\text{H}_2\text{SO}_4}}{P_{\text{T-H}_2\text{SO}_4} - P_{\text{H}_2\text{SO}_4}}$ is the levelized cost of electricity ($10^6$ dollars/MW·h) ($y_6 = 0$)

$C_{\text{T-H}_2\text{SO}_4}$ is the annualized cost of electricity without the sulfuric acid plant ($10^6$ dollars/year) ($y_6 = 1$)

$C_{\text{H}_2\text{SO}_4}$ is the annualized cost of a sulfuric acid plant ($10^6$ dollars/year) ($y_6 = 1$)

$P_{\text{T-H}_2\text{SO}_4}$ is the total power generated excluding power consumption in the sulfuric acid plant (MW·h) ($y_6 = 0$)

$P_{\text{H}_2\text{SO}_4}$ is the total power consumption in the sulfuric acid plant (MW·h) ($y_6 = 1$)

$x_1 = \eta_s$ is the desulfurization efficiency

$x_2 = t_a$ is the zinc ferrite cycle time (h)

$x_3 = R_{\text{Ca/S}}$ is the calcium-to-sulfur ratio

$E_{\text{SO}_2}$ is the total sulfur emissions (lbs $\text{SO}_2$/MM Btu)

This MINLP formulation does not have sufficient information in the master problem to produce the optimal solution. Therefore, the decision vector $\bar{x}$ and the equality constraints $h(\bar{x})$ are modified by introducing five additional (pseudo) variables $x_4$, $x_5$, $x_6$, $x_7$, and $x_8$:

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \rightarrow \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \hline x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}$$

and

$$H(\bar{x}) = \begin{bmatrix} \hline x_4 - C_{\text{T-H}_2\text{SO}_4} = 0 \\ x_5 - C_{\text{H}_2\text{SO}_4} = 0 \\ x_6 - P_{\text{T-H}_2\text{SO}_4} = 0 \\ x_7 - P_{H_2SO_4} = 0 \\ x_8 - E_{\text{SO}_2} = 0 \end{bmatrix}$$

Physically, the additional variables $x_4, ..., x_8$ are chosen to represent the flowsheet output variables $C_{\text{T-H}_2\text{SO}_4}$, $C_{\text{H}_2\text{SO}_4}$, $P_{\text{T-H}_2\text{SO}_4}$, $P_{\text{H}_2\text{SO}_4}$, and $E_{\text{SO}_2}$, respectively. The objective function, $C_{\text{elec}}$ can now be written in terms of the new decision variables $x_4$, $x_5$, $x_6$, and $x_8$ as follows:

$$\min_{x_1,x_2,x_3,x_4,x_5,x_6,x_7,x_8} Z = C_{\text{elec}} = \frac{x_4 + x_5}{x_6 - x_7}$$

subject to the emission constraint

$$x_8 \leq \frac{0.015 \text{ lb SO}_2}{10^6 \text{ Btu}}$$

For the above partitioned matrices, the Jacobian matrices can be evaluated as follows:

$$\frac{\partial Z(\bar{x})}{\partial \bar{x}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \hline \dfrac{1}{(x_6 - x_7)} \\ \dfrac{1}{(x_6 - x_7)} \\ \dfrac{-(x_4 + x_5)}{(x_6 - x_7)^2} \\ \dfrac{(x_4 + x_5)}{(x_6 - x_7)^2} \end{bmatrix}$$

$$\frac{\partial H(\bar{x})}{\partial \bar{x}} =$$

$$\left[\begin{array}{ccc|ccccc} \dfrac{-\partial C_{\text{T-H}_2\text{SO}_4}}{\partial x_1}_{\text{perturb}} & \dfrac{-\partial C_{\text{T-H}_2\text{SO}_4}}{\partial x_2}_{\text{perturb}} & \dfrac{-\partial C_{\text{T-H}_2\text{SO}_4}}{\partial x_3}_{\text{perturb}} & 1 & 0 & 0 & 0 & 0 \\ \dfrac{-\partial C_{\text{H}_2\text{SO}_4}}{\partial x_1}_{\text{perturb}} & \dfrac{-\partial C_{\text{H}_2\text{SO}_4}}{\partial x_2}_{\text{perturb}} & \dfrac{-\partial C_{\text{H}_2\text{SO}_4}}{\partial x_3}_{\text{perturb}} & 0 & 1 & 0 & 0 & 0 \\ \dfrac{-\partial P_{\text{T-H}_2\text{SO}_4}}{\partial x_1}_{\text{perturb}} & \dfrac{-\partial P_{\text{T-H}_2\text{SO}_4}}{\partial x_2}_{\text{perturb}} & \dfrac{-\partial P_{\text{T-H}_2\text{SO}_4}}{\partial x_3}_{\text{perturb}} & 0 & 0 & 1 & 0 & 0 \\ \dfrac{-\partial P_{\text{H}_2\text{SO}_4}}{\partial x_1}_{\text{perturb}} & \dfrac{-\partial P_{\text{H}_2\text{SO}_4}}{\partial x_2}_{\text{perturb}} & \dfrac{-\partial P_{\text{H}_2\text{SO}_4}}{\partial x_3}_{\text{perturb}} & 0 & 0 & 0 & 1 & 0 \\ \dfrac{-\partial E_{\text{SO}_2}}{\partial x_1}_{\text{perturb}} & \dfrac{-\partial E_{\text{SO}_2}}{\partial x_2}_{\text{perturb}} & \dfrac{-\partial E_{\text{SO}_2}}{\partial x_3}_{\text{perturb}} & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

where the left-hand side of the partition can be evaluated by perturbation and the right-hand side is the identity matrix. The new strategy avoided five calculation cycles (corresponding to five decision variable perturbations) of full simulation of the IGCC flowsheet at each NLP iteration (each simulation required approximately 90 s of CPU time on a VAX-3200). This reduced the computational time for the IGCC MINLP process synthesis problem by about 70%, which represents considerable savings in the computational efforts.

## 6. Conclusions

The implicit constraint problem encountered in MINLP process synthesis using sequential modular simulators is illustrated with a simple example considering two reactors. A strategy to circumvent this problem by adding pseudo decision variables, which are equated to the output variables of interest, was previously described. This strategy, however, increases the load on the NLP optimizer, mainly due to the derivative calculations based on the direct loop perturbations of the decision variables. The increased number of variables requires an increased number of perturbations and increased flowsheet simulations. In order to reduce the computational effort at the NLP optimization stage, a partitioning scheme was proposed. The decision variable matrix is divided into two parts: the real decision variables which affect the flowsheet simulation results and the newly added pseudo decision variables which do not affect the output of the flowsheet simulations. Because of this partitioning, the number of perturbations is reduced to the number of real decision variables, and the partial derivatives with respect to pseudo decision variables can be evaluated analytically. This scheme provides all the required information to the master problem without spending too much computational effort at the NLP optimization stage. The power of the partitioning scheme was demonstrated in the context of a large sized problem of an IGCC environmental control system design. Computational savings of about 70% were observed.

## Literature Cited


Diwekar, U. M.; Grossmann, I. E.; Rubin, E. S. An MINLP process synthesizer for a sequential modular simulator. *Ind. Eng. Chem. Res.* **1992a**, *31*, 313–322.

Diwekar, U. M.; Frey, H. C.; Rubin, E. S. Synthesizing optimal flowsheets: applications to IGCC system environmental control. *Ind. Eng. Chem. Res.* **1992b**, *31*, 1927–1936.

Floudas, C. S.; Aggrawal, A.; Ciric, A. R. Global optimum search for nonconvex NLP and MINLP problems. *Comput. Chem. Eng.* **1989**, *13*, 1117–1132.

Gill, P. E.; Murray, W.; Wright, M. H. *Practical Optimization*; Academic Press: London, 1981.

Kocis, G. R.; Grossmann, I. E. Relaxation strategy for the structural optimization of process flowsheets. *Ind. Eng. Chem. Res.* **1987**, *26*, 1869–1880.

Kocis, G. R.; Grossmann, I. E. Computational experience with DICOPT solving MINLP problems in process systems engineering. *Comput. Chem. Eng.* **1989a**, *13*, 307.

Kocis, G. R.; Grossmann, I. E. A modeling and decomposition strategy for MINLP optimization of process flowsheets. *Comput. Chem. Eng.* **1989b**, *13*, 797–819.

Kravanja, Z.; Grossmann, I. E. PROSYN-An MINLP synthesizer. *Comput. Chem. Eng.* **1990**, *14*, 1363.

Paules, G. E., IV; Floudas, C. A. APROS-algorithmic development methodology for discrete-continuous optimization problems. *Ops. Res. J.* **1989**, *37*, 902.

Salcedo, R. L. Solving nonconvex nonlinear programming and mixed-integer nonlinear programming problems with adaptive random search. *Ind. Eng. Chem. Res.* **1992**, *31*, 262–273.

Viswanathan, J.; Grossmann, I. E. A combined penalty function and outer approximation method for MINLP optimization. *Comput. Chem. Eng.* **1990**, *14*, 769.



*Received for review* November 23, 1992
*Revised manuscript received* April 26, 1993
*Accepted* May 3, 1993