An MINLP Process Synthesizer for a Sequential Modular Simulator

U. M. Diwekar,† **I. E. Grossmann,***‡ **and E. S. Rubin**†

Department of Engineering and Public Policy and Department of Chemical Engineering, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213

This paper deals with the development of a mixed integer nonlinear programming (MINLP) synthesizer for sequential modular simulators. Firstly, a variant of the outer approximation/equality relaxation/augmented penalty (OA/ER/AP) algorithm for MINLP problems is presented that makes use of Benders cuts in previous or subsequent iterations. An automatic process synthesis environment is then described for the public version of the ASPEN simulator using this algorithm, with the decomposition strategy by Kocis and Grossmann. The application of this new capability is demonstrated with several examples including the structural optimization of the hydrodealkylation of toluene process.

1. Introduction

Sequential modular chemical process simulators, such as FLOWTRAN, ASPEN, or PROCESS, have been widely used for the design of new processes, analysis of existing processes, etc. These flowsheeting programs contain very detailed models for calculating mass and energy balances as well as for sizing and costing. Over the past few years important advances have been made in flowsheet optimization with process simulators using nonlinear programming (NLP) techniques. Effective computational strategies based on the successive quadratic programming (SQP) algorithm (Biegler and Cuthrell, 1985; Han, 1977; Powell, 1978) have been developed. These strategies include the feasible and infeasible path optimization (Biegler and Hughes, 1983). In fact these advances have made optimization a standard computational option in most of the simulators nowadays.

A major goal in chemical process design is to synthesize flowsheet structures. Processes can be modeled and optimized using simulators with an NLP optimization capability. However, this optimization tool is restricted to flowsheets with fixed topology and therefore cannot be readily applied to process synthesis problems. One of the primary goals in process synthesis is to establish methodologies for determining optimal flowsheet configurations. The current state of such techniques involves (a) the heuristic approach which relies on intuition and engineering knowledge, (b) the physical insight approach which is based on exploiting basic physical principles, and (c) the optimization approach which uses the mathematical programming techniques. This paper deals with the development of a process synthesis prototype built around the sequential modular simulator ASPEN (public version) using the mathematical programming approach, which requires the solution of a mixed integer nonlinear programming (MINLP) problem (for general review see Grossmann (1990)).

Algorithms for solving MINLP optimization include the branch and bound method (Beale, 1977; Gupta, 1980), the generalized Benders decomposition (Benders, 1962; Geoffrion, 1972), and the outer approximation (OA) method (Duran and Grossmann, 1986). The OA and GBD algorithms are in general more efficient than the branch and bound method. The OA algorithm requires fewer iterations than GBD but involves the solution of a larger master problem. Another difficulty with these algorithms is that they require the functions to satisfy convexity conditions to guarantee convergence to the global optimum. Recent variants of the OA method include the outer approximation/equality relaxation (OA/ER) strategy of Kocis and Grossmann (1987) for handling nonlinear equations and the augmented penalty OA/ER algorithm of Viswanathan and Grossmann (1990) for reducing the effect of nonconvexities in the master problem. Also the extension of GBD using a partioning variable strategy by Floudas et al. (1989) shows improved results for nonconvex functions. Since the direct application of MINLP algorithm to flowsheet synthesis poses serious difficulties, Kocis and Grossmann (1989) developed a modeling and decomposition strategy that requires only the NLP optimization of the existing flowsheet at each iteration, avoiding the need of handling units with zero flows. Recently, Kravanja and Grossmann (1990) implemented this strategy in PROSYN, within an equation-oriented environment in which the simultaneous optimization and heat integration was also considered. This strategy, however, has not been implemented in sequential modular simulators. It should be mentioned that Harsh et al. (1989) have used the OA algorithm in FLOWTRAN for the optimal retrofit design but the topology of the flowsheet was fixed. Caracotsios and Petrelli (1989) have built an MINLP environment which can be interfaced with the ASPEN simulator. Applications to the synthesis of complex chemical processes have not been reported in their research.

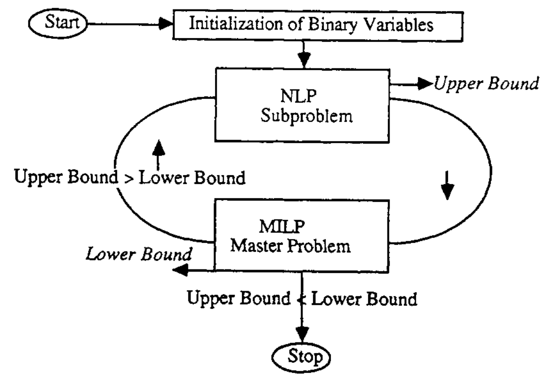


Figure 1. Main steps in GBD and OA algorithms.

In this paper the process synthesis capability that has been implemented in the ASPEN (public version) simulator is described. For the MINLP optimization, a variant of the GBD and OA algorithm is proposed which is relatively easy to implement in process simulators. In addition the decomposition strategy by Kocis and Grossmann

* Author to whom correspondence should be addressed.
† Department of Engineering and Public Policy.
‡ Department of Chemical Engineering.

0888-5885/92/2631-0313$03.00/0

(1989) is used to circumvent the problem of zero flows. Several examples are presented including a restricted version of the synthesis of the HDA process (Douglas, 1988).

2. MINLP Techniques

Process synthesis involves defining a search space or a superstructure of candidate flowsheet structures that may be based on preliminary screening (Grossmann, 1990). his superstructure can then be modeled as an MINLP problem of the form

MINLP:

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

subject to

$$h(x) = 0$$

$$By + g(x) \leq 0$$

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

where

$Y = \{y|Ay \leq a, y \{0,1\}^{m}\}; \quad X = \{x|x^{L} \leq x \leq x^{U}, x \in R^{n}\}$

The continuous variables x represent flows, operating conditions, and design variables. The binary variables y denote the potential existence of process units.

The GBD and OA algorithms for solving the above MINLP consist of solving at each major iteration and NLP subproblem (with all 0–1 variables fixed) and an MILP master problem as shown in Figure 1. The NLP subproblems have the role of optimizing the continuous variables and provide an upper bound to the optimal MINLP solution. The MILP master problem has the role of predicting a lower bound to the MINLP as well as new 0–1 variable values for each major iteration. The predicted lower bounds increase monotonically as the cycle of major iterations proceeds, and the search is terminated when the predicted lower bound coincides or exceeds the current upper bound.

The main difference between GBD and the OA method lies in the definition of the MILP master problem. The master problem in GBD is a dual representation of the continuous space, while the master problem in OA is given by a primal approximation.

2.1. Master Problems. 2.1.1. GBD. Based on the solution of k NLP subproblems with fixed y^{k}, $k = 1, ..., K$, the master problem of GBD is given by

GBD:

$$Z_{\mathrm{GBD}}^{k} = \min_{y,\alpha_{\mathrm{GBD}}} \alpha_{\mathrm{GBD}}$$

subject to

$$\alpha_{\mathrm{GBD}} \geq c^{T}y + f(x^{k}) + \sum_{j=1}^{t1} \mu_{j}^{k}[g_{j}(x^{k}) + b_{j}y] \qquad k = 1, 2, ..., K$$

$$y \in Y; \quad \alpha_{\mathrm{GBD}} \in R^{1}$$

where $\quad Y = \{y|Ay \leq a, y \in \{0,1\}^{m}\}$

where k represents the iteration counter and μ_{j}^{k}'s are the Lagrange multipliers of the inequality constraints. The master problem of GBD contains only the 0–1 variables and the Lagrangian cut in the 0–1 space. Since the master problem does not involve continuous variables, it predicts weaker lower bounds.

2.1.2. OA Algorithm and Its Extensions. The master problem for OA is given by

OA:

$$Z_{\mathrm{OA}}^{K} = \min_{y,x,\alpha_{\mathrm{OA}}} \alpha_{\mathrm{OA}}$$

subject to

$$\alpha_{\mathrm{OA}} \geq c^{T}y + f(x^{k}) + \nabla f(x^{k})^{T}(x - x^{k})$$

$$By + g(x^{k}) + \nabla g(x^{k})^{T}(x - x^{k}) \leq 0 \qquad k = 1, 2, ..., K$$

$$y \in Y; \quad x \in X; \quad \alpha_{\mathrm{OA}} \in \mathrm{R}^{1}$$

where $Y = \{y|Ay \leq a, y \in \{0,1\}^{m}\};$
$X = \{x|x^{L} \leq x \leq x^{U}, x \in R^{n}\}$

As the iterations proceed, the master problem accumulates constraints in the $X-Y$ space generating richer information, so that OA converges in fewer iterations than the GBD. Extension of this algorithm for handling nonlinear equality constraints is the OA/ER algorithm of Kocis and Grossmann (1987). Here, linearizations of the equations at the solution of the NLP subproblem k are added to the master problem (OA) by relaxing them according to the sign of the Lagrange multiplier; that is

ER:

$$T^{k}[\nabla h(x^{k})^{T}(x - x^{k})] \leq 0$$

where $T^{k} = [t_{ii}^{k}]$

$$t_{ii}^{k} = \begin{cases} 1 \text{ if } \lambda_{i}^{k} > 0 \\ 0 \text{ if } \lambda_{i}^{k} = 0 \\ -1 \text{ if } \lambda_{i}^{k} < 0 \end{cases}$$

As shown by Kocis and Grossmann (1987), sufficient conditions for global optimality with the OA/ER algorithm require convexity in the objective function and inequalities and quasi-convexity of the relaxed nonlinear equations. Nonconvexities can be treated using the recent variant of this algorithm (Viswanathan and Grossmann, 1990) that makes use of augmented penalty function in the master problem. The algorithm starts with solution of the relaxed NLP. The master problem contains the slack variables and is given by

OA/ER/AP:

$$Z_{\mathrm{AP}}^{K} = \min_{y,x,\alpha_{\mathrm{AP}},p^{k},q^{k},s^{k}} \alpha_{\mathrm{AP}} + \sum_{k=1}^{K}(w_{q}^{k})^{T}q^{k} + \sum_{k=1}^{K}(w_{p}^{k})^{T}p^{k} + \sum_{k=1}^{K}(w_{s}^{k})^{T}s^{k}$$

subject to

$$\alpha_{\mathrm{AP}} \geq c^{T}y + f(x^{k}) + \nabla f(x^{k})^{T}(x - x^{k}) - q^{k}$$

$$T^{k}[\nabla h\,(x^{k})^{T}(x - x^{k})] \leq p^{k}$$

$$By + g(x^{k}) + \nabla g\,(x^{k})^{T}(x - x^{k}) \leq s^{k} \qquad k = 1, 2, ..., K$$

$$y \in Y; \quad x \in X; \quad \alpha_{\mathrm{AP}} \in R^{1}$$

where $Y = \{y|Ay \leq a, y \in \{0,1\}^{m}\};$
$X = \{x|x^{L} \leq x \leq x^{U}, x \in R^{n}\}$

where q^k, p^k, and s^k are the slack variables with corresponding large finite weights w_q^k, w_p^k, and w_s^k.

The OA/ER/AP algorithm proceeds until there is no decrease in the NLP solution. Experience with this algorithm has shown a high degree of robustness with nonconvex problems.

2.1.3. GBD/OA/ER/AP; A New Variant. Although the OA/ER/AP algorithm is computationally efficient and has shown a high degree of robustness, it requires solution of the relaxed NLP at the initial point. In the context of a process flowsheet this involves optimization of the entire superstructure which can be computationally quite expensive. Another problem with this algorithm is that the master problem grows in size as the iterations proceed which calls for major restructuring of the master problem at each iteration. In this paper we propose a hybrid method to overcome these problems and to make the algorithm easier to implement. The basic idea is to apply the OA/ER/AP at the current iteration k, while previous iterations are stored in the form of Benders cuts. The master problem is then given by

GBD/OA/ER/AP1:

$$Z_{\mathrm{GOEA}}^K = \min_{y,x,\alpha_{\mathrm{GOEA}}} \alpha_{\mathrm{GOEA}}$$

subject to

$$\alpha_{\mathrm{GOEA}} \geq c^{\mathrm{T}}y + f(x^K) + \nabla f(x^K)^{\mathrm{T}}(x - x^K) + (w_p^K)^{\mathrm{T}}p^K + (w_s^K)^{\mathrm{T}}s^K$$

$$T^K[\nabla h(x^K)^{\mathrm{T}}(x - x^K)] \leq p^K$$

$$By + g(x^K) + \nabla g(x^K)^{\mathrm{T}}(x - x^K) \leq s^K$$

$$\alpha_{\mathrm{GOEA}} \geq c^{\mathrm{T}}y + f(x^k) + \nabla f(x^k)^{\mathrm{T}}(x - x^k) + \sum_{j=1}^{t1}\mu_j[b_jy + g_j(x^k) + \nabla g_j(x^k)^{\mathrm{T}}(x - x^k)] \qquad k = 1, 2, ..., K - 1$$

$$\sum_{j\in B^k} y_j^k - \sum_{i\in N^k} y_i^k \leq |B^k| - 1 \qquad k = 1, 2, ..., K$$

$$y \in Y; \quad x \in X; \quad \alpha_{\mathrm{GOEA}} \in R^1$$

where $Y = \{y|Ay \leq a, y \in \{0,1\}^m\}$;
$X = \{x|x^{\mathrm{L}} \leq x \leq x^{\mathrm{U}}, x \in R^n\}$

Alternatively one can keep the first iteration in the form of OA/ER/AP and the successive iterations treated in the form of GBD constraints as shown below (GBD/OA/ER/AP2). This formulation is especially designed for the decomposition strategy (section 2.2), where the first iteration constraints are richer in information.

GBD/OA/ER/AP2:

$$Z_{\mathrm{GOEA}}^k = \min_{y,x,\alpha_{\mathrm{GOEA}}} \alpha_{\mathrm{GOEA}}$$

subject to

$$\alpha_{\mathrm{GOEA}} \geq c^{\mathrm{T}}y + f(x^1) + \nabla f(x^1)^{\mathrm{T}}(x - x^1) + (w_p^1)^{\mathrm{T}}p^1 + (w_s^1)^{\mathrm{T}}s^1$$

$$T^1[\nabla h(x^1)^{\mathrm{T}}(x - x^1)] \leq p^1$$

$$By + g(x^1) + \nabla g(x^1)^{\mathrm{T}}(x - x^1) \leq s^k$$

$$\alpha_{\mathrm{GOEA}} \geq c^{\mathrm{T}}y + f(x^k) + \nabla f(x^k)^{\mathrm{T}}(x - x^k) + \sum_{j=1}^{t1}\mu_j[b_jy + g_j(x^k) + \nabla g_j(x^k)^{\mathrm{T}}(x - x^k)] \qquad k = 2, 3, ..., K$$

$$\sum_{j\in B^k} y_j^k - \sum_{i\in N^k} y_i^k \leq |B^k| - 1 \qquad k = 1, 2, ..., K$$

$$y \in Y; \quad x \in X; \quad \alpha_{\mathrm{GOEA}} \in R^1$$

where $Y = \{y|Ay \leq a, y\{0,1\}^m\}$;
$X = \{x|x^{\mathrm{L}} \leq x \leq x^{\mathrm{U}}, x \in R^n\}$

The solution strategy is similar to the OA/ER/AP algorithm except that the solution does not start with the relaxed NLP, but for a fixed value of the 0–1 variables. This method has shown similar robustness as observed in the OA/ER/AP algorithm and is easier to implement. It should also be noted that for the convex case, where the slack variables are driven to zero (see Viswanathan and Grossmann (1990)), the master problem GBD/OA/ER/AP1 does not necessarily yield nondecreasing lower bounds. However convergence to the optimum can still be guaranteed when the bound predicted by the master problem is greater or equal than the best upper bound.

An Illustrative Example. The GBD/OA/ER/AP algorithm is illustrated by solving the following example problem:

EX1:

$$\min y1 + 1.5(y2) + 0.5(y3) + x11 + x12$$

subject to

$$x11 - x1^2 = 0$$

$$x12 - x2^2 = 0$$

$$x13 - x2 - (x1 - 2)^2 = 0$$

$$x13 \geq 0$$

$$2(y1) - x1 \leq 0$$

$$1 - y1 - x1 \leq 0$$

$$3(y3) - x1 - x2 \leq 0$$

$$y1 + y2 + y3 \geq 1$$

$$y1, y2, y3 = 0, 1$$

Table I presents the comparison of the step by step results obtained using GBD/OA/ER/AP1 and GBD/OA/ER/AP2 with the OA/ER/AP algorithm. The termination criterion in the three cases is based on the increase of the objective function in the NLP.

2.2. Decomposition Strategy. Although one can directly use either of the two above versions of the above stated MINLP algorithm for the optimization of superstructures of flowsheets, it is clear that, in order to increase the reliability and the efficiency of the solution procedure, one ought to recognize the special structure and properties that characterize the optimal synthesis of process flowsheets. The main difficulty which is encountered here is having to optimize "dry units" with zero flow, which are temporarily turned off in the superstructure. One way to handle this problem is to use very small flows instead of zero flows (Kocis and Grossmann, 1987; Caracotsios and Petrellis, 1989). However, this is computationally ineffi-

Table I. Comparison of GBD/OA/ER/AP with OA/ER/AP Algorithms (EX1)[a]

major step	major iteration	objective function OA/ER/AP	GBD/OA/ER/AP1	GBD/OA/ER/AP2	binary variables OA/ER/AP			GBD/OA/ER/AP1			GBC/OA/ER/AP2		
NLP	1	2.5353	6.5000	6.5000	0.377	0	0.623	0	1	1	0	1	1
MILP	1	3.3956	3.0000	3.0000	0	1	0	0	1	0	0	1	0
NLP	2	**3.5000**	**3.5000**	**3.5000**	**0**	1	**0**	**0**	1	**0**	**0**	1	**0**
MILP	2	3.7715	5.4000	4.7000	1	0	0	1	0	0	1	0	0
NLP	3	5.0000	5.0000	5.0000	1	0	0	1	0	0	1	0	0

[a] Optimal solution: $x1 = 1.0$, $x2 = 1.0$, $x11 = 1.0$, $x12 = 1.0$, $x13 = 0.0$, objective function = 3.5.

cient since one has to solve an NLP problem for the entire superstructure which can also lead to numerical difficulties due to the very small flows in the units; for instance, linearizations can result in very poor approximations to the nonlinear functions. Lastly, the likelihood of getting trapped in multiple optima is greater in the larger NLP problem. The decomposition strategy of Kocis and Grossmann (1989) has the important feature that the NLP optimization is only required for the flowsheet with existing units at each iteration. Information to the master problem on the nonexisting units is provided through a Lagrangian suboptimization procedure which enforces nonzero flows in these units.

The decomposition scheme is similar in nature to multilevel optimization methods that use Lagrange multipliers to decompose the separable problems. The input variables in the problem can be separated into two groups; the variables associated with existing units and the variables associated with the nonexisting units. The initial problem is formulated using the existing units and the subproblem (subproblems) using nonexisting units. Material balance equations at interconnection nodes are used to relate the initial problem to the subproblem (for details see Kocis and Grossmann (1989)). The suboptimization procedure is used to get good points for linearizations and hence is performed only at iteration 1. Since iteration 1 has all the information of the superstructure, GBD/OA/ER/AP2 formulation is used to solve this problem. The following example of two reactors will illustrate the decomposition strategy.

Two-Reactor Problem. Consider the problem by Kocis and Grossmann (1989) for selecting from among two candidate reactors to minimize the cost of producing a desired product (see Figure 2). The MINLP formulation is

TWO-REA:

$$\min 7.5(y1) + 5.5(y2) + 7(v1) + 6(v2) + 5x$$

subject to

$$z1 = 0.9[1 - \exp(-0.5(v1))]x1$$

$$z2 = 0.8[1 - \exp(-0.4(v2))]x2$$

$$x1 + x2 - x = 0$$

$$z1 + z2 = 10$$

$$v1 \le 10(y1)$$

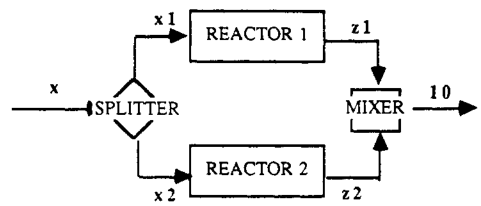
$$v2 \le 10(y2)$$

$$x1 \le 20(y1)$$

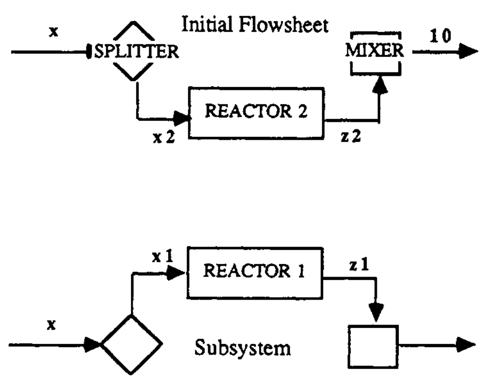
$$x2 \le 20(y2)$$

$$y1 + y2 = 1$$

$$y1, y2 = 0, 1; \quad x1, x2, z1, z2, v1, v2 \ge 0$$



a) Superstructure



b) Initial flowsheet and the subsystem

Figure 2. Superstructure for the two-reactor problem.

The binary variables $y1$ and $y2$ denote existence of reactors 1 and 2. The material balance constraints for interconnection nodes include splitting of input flows $x1$ and $x2$ and mixing of outlets $z1$ and $z2$. This problem was solved using GBD/OA/ER/AP2 algorithm with the decomposition scheme shown in Figure 2b. The splitter, the mixer, and the reactor were selected as the initial flowsheet while the remaining units are suboptimized.

I. First NLP

(a) The NLP optimization of the initial flowsheet is given by

$$\min 5.5 + 6(v2) + 5x$$

subject to

$$z2 = 0.8[1 - \exp(-0.4(v2))]x2$$

$$x2 - x = 0$$

$$z2 = 10$$

$$v2 \le 10$$

$$x2 \le 20$$

The solution to this NLP is $Z = 107.376$ at $x2 = x = 15$ and $v2 = 4.479$, and the Lagrange multiplier for the mixer mass balance ($z2 = 10$) is -7.5. Linearizations for the nonlinear constraints corresponding to the initial flowsheet result in

$$z2 - 0.666(x2) - 0.800(v2) + 3.584 - p2 \le 0$$

(b) The NLP suboptimization problem for the nonexistent unit is then given by

$$\min 7.5 + 7(v1) + 5(x1) - 7.5(z1)$$

subject to

$$z1 = 0.9[1 - \exp(0.5(v1))]x1$$

$$x1 - 15 = 0$$

$$v1 \le 10$$

$$x1 \le 20$$

Note that the material balance equation for inlet to the interconnection node is replaced by the constraint which fixes the inlet flow rate equal to the inlet flow rate calculated at the optimal value of the initial flowsheet ($x1 - 15 = 0$), and the interconnection node outlet material balance constraint is combined with the objective function using the Lagrange multiplier obtained at the initial flowsheet optimization. The solution yields $Z = 22.950$ at $z1 = 11.633$ and $v1 = 3.957$. The linear approximation to the corresponding constraint results in

$$z1 - 0.776(x1) - 0.933(v1) - 3.693 - p1 \le 0$$

II. First MILP Master Problem

The formulation of this problem is similar to TWO-REA with the only difference that the two nonlinear equations are replaced by the two linear approximations and the integer constraint ($y2 - y1 \le 0$) is also included. The solution is 95.64 ($\alpha - \sum_{j=1}^{2} W_j p_j$) at $y1 = 1$ and $y2 = 0$ indicating the existence of the first reactor and nonexistence of the second one.

III. Second NLP Subproblem

Continuous optimization of the NLP problem (only for the selected flowsheet) is performed. The solution yields $z = 99.240$.

IV. Second MILP Master Problem

Adding the following additional constraints

$$\alpha \ge 7.5(y1) + 5.5(y2) + 7(v1) + 6(v2) + 5x - 6.7(z2) + 6(v2) + 5(x2)$$

$$y1 - y2 \le 0$$

leads to an infeasible solution, and hence, the procedure terminates. Thus the optimal of second NLP represents the global optimum. The process synthesis environment in ASPEN uses the GBD/OA/ER/AP2 algorithm and the decomposition strategy by Kocis and Grossmann (1989). The Appendix describes the steps involved in this procedure.

3. Implementation in ASPEN

3.1. Process Synthesis Capability. The MINLP process synthesis capability in the public version of ASPEN is based on ZOOM (Marsten, 1986), the mixed integer linear programming (MILP) solver, and on SCOPT (Lang and Biegler, 1987), the nonlinear programming (NLP) solver. The overall structure of the process synthesis environment is shown in Figure 3. Optimization of the MINLP process synthesis problem is decomposed into continuous optimization of NLP problems at fixed choice of the binary variables y and discrete optimization through the MILP master problem. The MILP solver (Master) and the NLP optimizer are implemented in ASPEN as unit operation blocks and can be executed easily with the ASPEN process unit blocks.

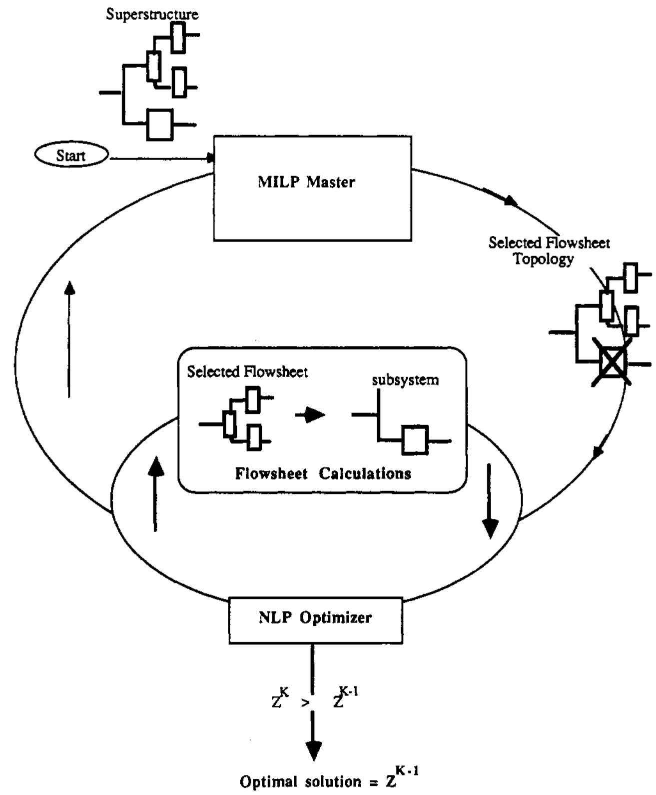


Figure 3. Process synthesizer for the ASPEN simulator.

Firstly a superstructure is postulated which has embedded alternative flowsheet structures. The superstructure is then modeled as an MINLP problem. Secondly the superstructure is decomposed into the initial flowsheet and subsystems of nonexisting units that are to be suboptimized with a Lagrangian scheme, in order to provide a linear approximation of the entire superstructure in the master problem. For more information on decomposition of the superstructure readers are advised to refer to the paper by Kravanja and Grossmann (1990).

The process synthesis environment in ASPEN consists of the Master block, the NLP optimizer, and the entire superstructure. The initialization of the continuous and binary variables is done in the ASPEN input file. At this stage the scheme is translated into the initial or the selected flowsheet and subsystems. NLP optimization of the selected flowsheet is the first step in the inner loop. The information in the Lagrangian multipliers is passed from the initial flowsheet to the subsystems to carry out their optimization. The inner loop results in the objective function value from the selected flowsheet optimization and in gradients and Lagrange multipliers from the selected flowsheet and subsystems. This information is passed to the Master block which internally modifies the master problem using the equality relaxation strategy and the information from the inner loop. The solution of the master problem results in a new flowsheet structure. From this point on, the NLP optimization is only performed on the new selected flowsheet. The iteration stops when there is no improvement (decrease) in the objective function value.

3.2. Input Structure. The input structure of the process synthesizer follows ASPEN's keyword input language for simulation and ZOOM's XML language for the

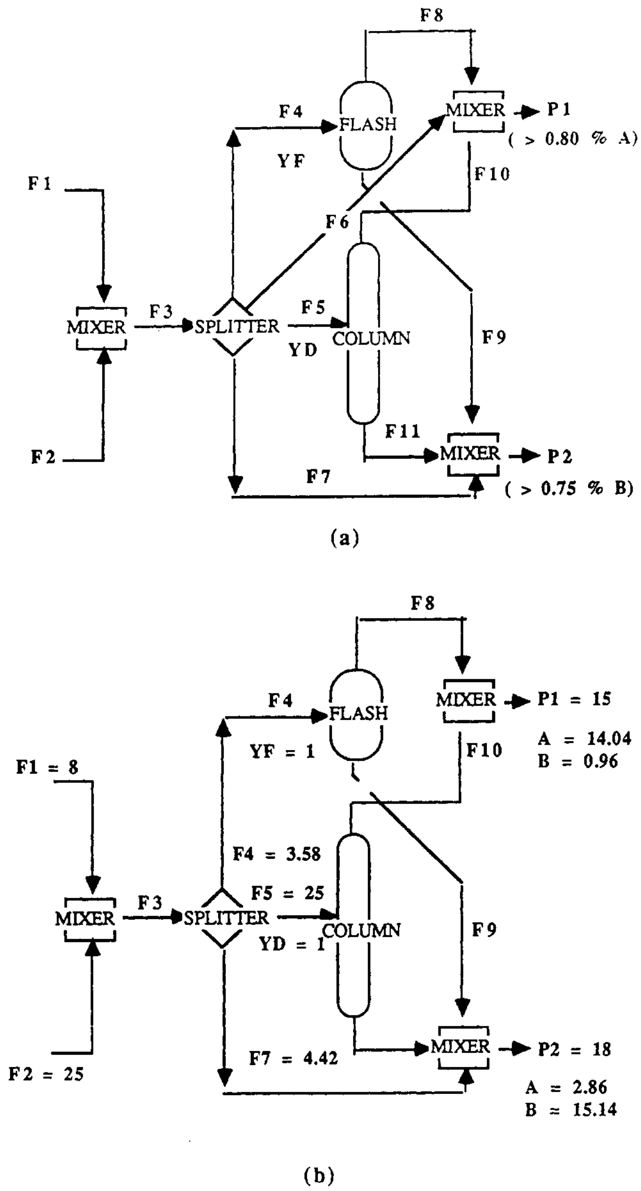


Figure 4. Superstructure (a) and optimal solution (b) for the separation synthesis problem.

MILP master problem. This type of environment is very easy to code and the information is transparent to the user. The algebraic equation oriented language for the MILP master problem (XML language) is simpler than providing the data and constraint equations through a Fortran subroutine. It internally generates the MPS files which can be used in any other MILP solver. The ASPEN input file is keyword driven and contains all the information about the topology of the superstructure, binary variables defining the selected flowsheet (subsystems), along with the input data for the process.

The transfer of gradient information and linearizations of the constraints is carried out internally. The information is stored in arrays and used in calculation directly instead of generating new MPS files as in ACCOPT (Carcotsios and Petrelli, 1989) or in FLOWTRAN for the retrofit strategy by Harsh et al. (1989). This saves on the unnecessary input/output.

3.3. Implicit Constraints Problem. In an equation oriented environment like PROSYN and GAMS, the nonlinear equality constraints are specified explicitly. For sequential modular simulators like ASPEN most of the constraints are implicit. This includes the black box relation between the output variables, which are part of the objective function or constraints in master problem and the input decision variables. The master problem needs the linearizations of these constraints. To circumvent the problem of singularities with implicit constraint linearizations, the following strategy is used in the NLP optimization. The continuous variable vector is separated into two categories, input variables and output variables, which results in

Table II. Results for the Separation Synthesis Problem

initial point	iteration	NLP	MILP	optimum	CPU time[a]
YD = 0, YF = 0	1	0	(0, 1)		12.62
	2	478	(1, 0)		
	3	482	(1, 1)		
	4	510	infeasible	510	
YD = 0, YF = 1	1	478	(1, 0)		10.75
	2	482	(1, 1)		
	3	510	infeasible	510	
YD = 1, YF = 0	1	482	(1, 1)		8.86
	2	510	(0, 1)		
	3	478		510	
YD = 1, YF = 1	1	510	(0, 1)		6.46
	2	478		510	

[a] Seconds, total ASPEN run on VAX 3200.

$$z^k_{\mathrm{NLP}} = \min_u f(u,v)$$

subject to

$$g(u) + By \le 0$$

$$h_1(u,v) = 0$$

$$u \in U \quad \text{and} \quad v = \phi(u) \quad \text{(implicit function)}$$

where u are input variables and v are output variables. The NLP problem is transformed to

IMP-CONS:

$$z^k_{\mathrm{NLP}} = \min_{u,v} f(u,v)$$

subject to

$$g(u) + by \le 0$$

$$h_1(u,v) = 0$$

$$h_2(u,v) = v - \phi(u)$$

$$u \in U, \quad v \in V$$

where $h_2(u,v)$ are additional constraints added to the NLP problem.

4. Examples

Firstly the GBD/OA/ER/AP approach presented in this paper is illustrated through the separation sequence problem described by Kocis and Grossmann (1989). These authors used special models for splitters and mixers to handle the nonconvexities in this problem. With the GBD/OA/ER/AP approach one need not use the special modeling scheme, since the nonconvexities are treated with the augmented penalty in the master problem. This feature is very useful for sequential modular simulators, as this strategy does not call for changes in the modules of the simulator. Application of the newly developed process synthesis capability for synthesis problems is also demonstrated with a restricted version of the synthesis of a process for the hydrodealkylation of toluene to produce benzene (Douglas, 1988).

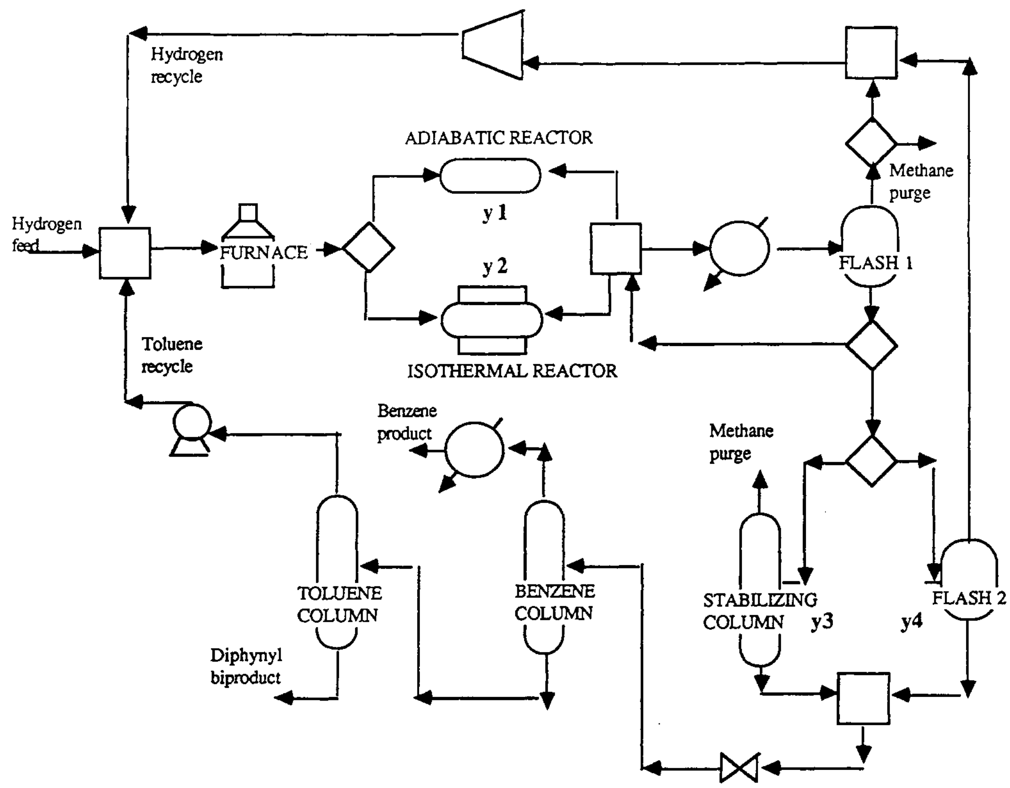


Figure 5. Superstructure for the HDA process synthesis problem.

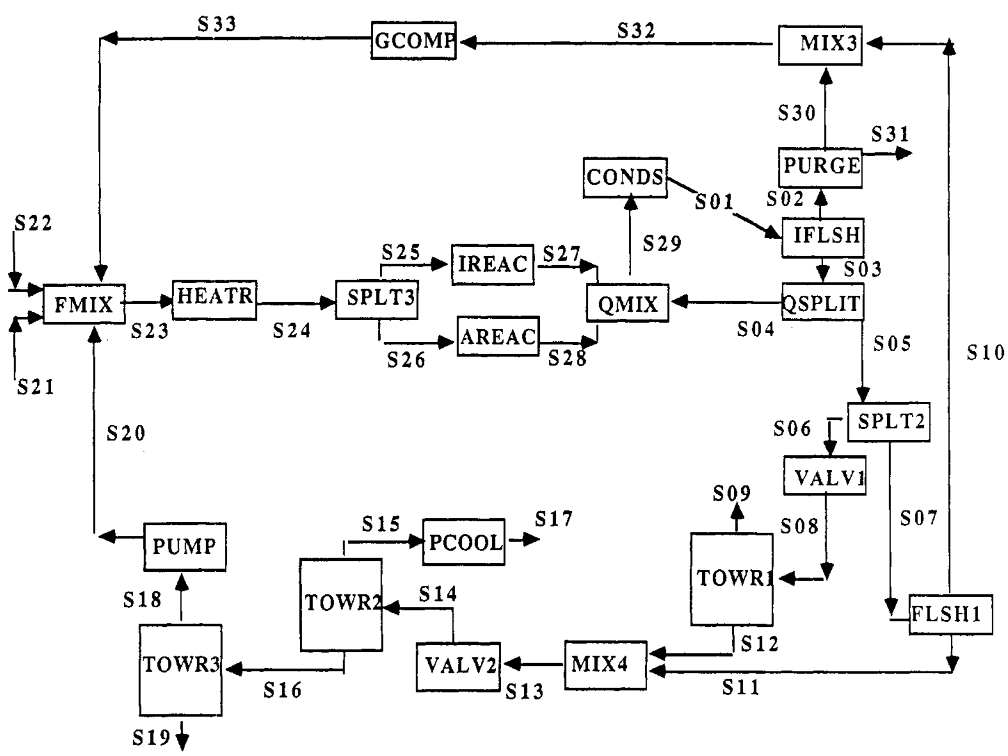


Figure 6. ASPEN representation of the superstructure for the HDA process synthesis problem.

4.1. Separation Scheme Synthesis. Consider the problem in Figure 4 of selecting the optimal separation scheme to be used to separate two components (A and B) which are available in feed streams F1 and F2. The compositions of these streams are 55% A, 45% B and 50% A, 50% B, respectively, and the desired product streams are P1 and P2. Purity specifications are a minimum of 80% A in product P1 and a minimum of 75% B in P2. Upper bounds are specified for the amounts of these products. Hence, there is the possibility of producing as much as these amounts or, at the other extreme, not producing any product if the separation scheme proves to be unprofitable.

Figure 4a is the superstructure of alternative separation schemes which can be used to deliver the desired product streams. Alternatives embedded in this superstructure include flash separation with blending, distillation with blending, flash separation and distillation in parallel, or the elimination of the complete separation process. The

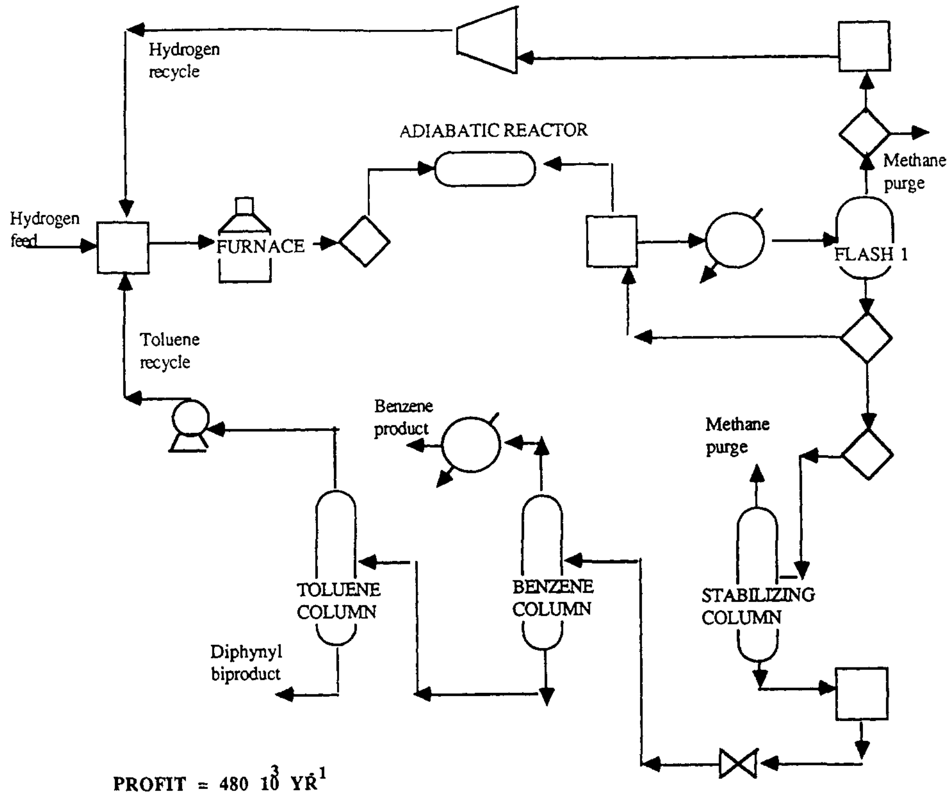


Figure 7. Optimal flowsheet structure for the HDA process synthesis problem for stringent purity requirements (>99% benzene).

binary variables associated with these decisions are also shown in Figure 4a. Since the components used in this example are arbitrary, we simulated this example by writing a small Fortran model in ASPEN. This model is a black box relating the output to the input variables and is in line with the simulator philosophy. The process synthesis capability is used to optimize this problem, and the results are presented in Table II. It can be seen that in all the cases (with different initial conditions) the solution reaches the global optimum YD = 1, YF = 1 with objective 510. It should be noted that Kocis and Grossmann (1989) required a special strategy for modeling the splitters and the mixers which is not the case here.

4.2. The HDA Process. The process chosen is the hydrodealkylation (HDA) of toluene process to produce benzene described extensively in Douglas (1988). The problem addressed is the selection of the flowsheet structure and operating conditions that maximize profit. Given a flowsheet superstructure of alternatives, this problem can be formulated as an MINLP problem as shown by Kocis and Grossmann (1989). In this paper, however, we will be using a somewhat smaller superstructure and different input conditions.

The superstructure selected for this problem is shown in Figure 5. The desired reaction in the HDA process is *toluene + hydrogen → benzene + methane*. An undesired reversible reaction occurs: 2*benzene → diphenyl + hydrogen*. The conditions for these gas-phase reactions are a pressure of 3.45 MPa and a temperature between 1150 and 1300 °F. At lower temperatures, the toluene reaction is too slow, and at higher temperatures hydrocracking takes place. Also a ratio of at least 5:1 moles of hydrogen to moles of aromatics is required to prevent coking. The kinetics and the design equations for the reactor are given in Douglas (1988).

A hydrogen raw material stream is available at a purity of 95% (the remaining 5% is methane). A toluene fresh feed stream is also available. These feed streams are combined with the recycle hydrogen and toluene streams which must be heated before being fed to the reactor. The exothermic reaction can be carried out in a plug flow reactor operating either adiabatically ($y1 = 1$) or isothermally ($y2 = 1$). The reactor product stream will contain unreacted hydrogen and toluene as well as the desired benzene product and undesired diphenyl and methane. This stream must be quenched immediately to prevent coking in the heat exchanger. The stream will be cooled further in order to condense the aromatics which will then be separated from the noncondensible hydrogen and methane in a flash separator.

The vapor stream leaving the flash separator contains valuable hydrogen which can be recycled. A portion of the flash separator liquid stream is used to quench the reactor product stream, and the remainder is sent to the liquid separation system. Since this stream may contain hydrogen and methane, it is necessary to remove these components using a stabilizing column ($y3 = 1$) or, alternatively, a second flash separator ($y4 = 1$) operating at lower pressure than the first flash. Having removed the hydrogen and methane, the liquid streams contain benzene, toluene, and diphenyl. The benzene product is specified to be at least 99% pure, at a production rate of 1.2 kmol/s. A distillation column is required to yield a product stream of this purity. The bottom stream leaving the benzene column contains primarily toluene, with a small amount of diphenyl and possibly some benzene. Prior to recycling the unreacted toluene, diphenyl should be removed by a column. This process is modeled using the ASPEN simulator. Figure 6 shows the ASPEN representation of the superstructure. Simplifications were made wherever necessary using guidelines and data from Douglas (1988).

The objective function is the maximization of the annualized profit. The cost model is represented by linear fixed-charge costs and the data are given in Table III. The decomposition strategy is used to solve this problem.

The resulting MINLP optimization problem contains

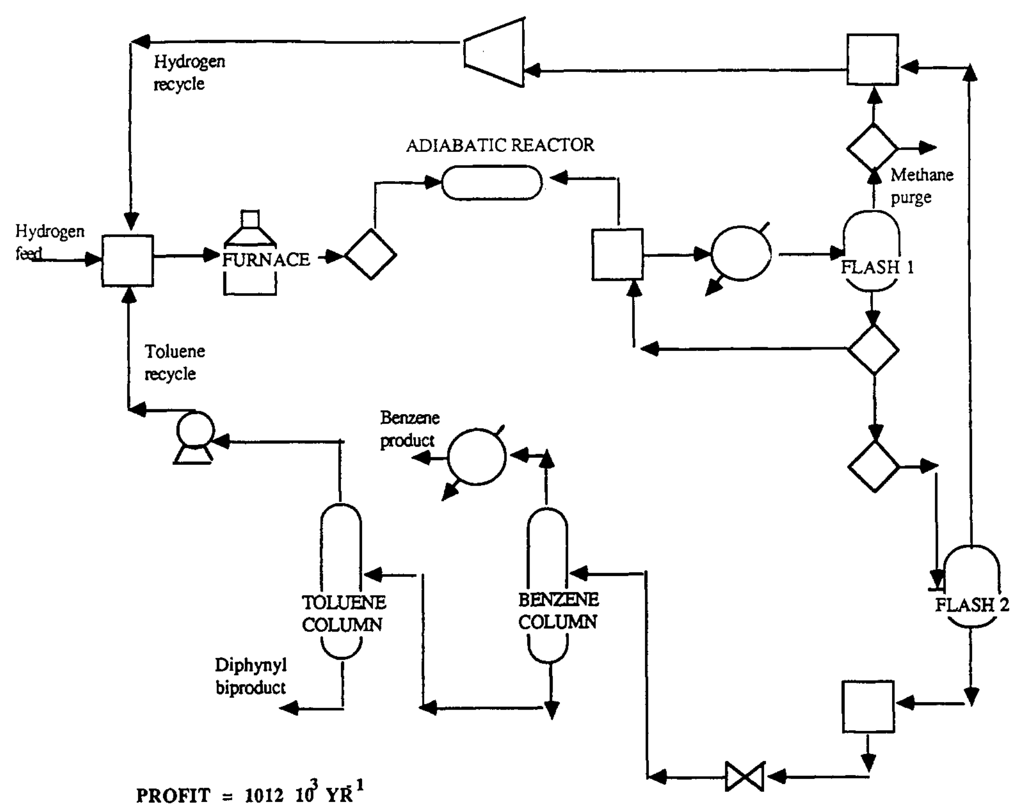


Figure 8. Optimal flowsheet structure for the HDA process synthesis problem for less stringent purity requirements (>95%).

Table III. Cost Data for HDA Problem

feedstock or product/byproduct		costs/price, \$/kg-mol
hydrogen feed	95% hydrogen 5% methane	2.50
toluene feed	100% toluene	14.00
benzene product	≥95% benzene	19.90
diphenyl product		11.84
hydrogen purge	(heating value)	1.08
methane purge	(heating value)	3.37
utilities		**costs**
electricity		\$0.04 kW/h
heating (steam)		\$8.0 × 10⁶/kJ
cooling (water)		\$0.7 × 10⁶/kJ
fuel		\$4.0 × 10⁶/kJ
investment costs, \$10³ year⁻¹	**fixed-charge cost**	**linear coefficient**
compressor	7.155	0.815 × bhp (kW)
stabilizing column	1.126	0.375 × no. of trays
benzene column	16.3	1.55 × no. of trays
toluene column	3.90	1.12 × no. of trays
furnace	6.20	1172 × heat duty (10⁶ kJ/year)
reactor (adiabatic)	74.3	1.257 × reactor vol (m³)
reactor (isothermal)	92.875	1.571 × reactor vol (m³)

4 0–1 variables, 14 continuous decision variables, and 11 additional constraints (constraints made explicit). The optimal flowsheet structure is shown in Figure 7. It is interesting to note that if the constraint on purity is made less stringent i.e., purity is more than 95% instead of 99%, the optimal flowsheet structure contains the flash unit instead of the stabilizing column (Figure 8). In most of the cases with different initial values the global optimum is found in three or less than three NLP iterations and two or less MILP iterations.

5. Conclusions

This paper has described the implementation of an MINLP process synthesizer in the ASPEN process simulator. A variant of the GBD and the OA algorithms for solution of MINLP problems has been presented. The proposed algorithm makes use of the OA/ER/AP approximation for one iteration and of GBD cuts for subsequent iterations which allows a convenient implementation in a sequential modular simulation environment.

The algorithm has been implemented along with the decomposition scheme of Kocis and Grossmann (1989) in the sequential modular simulator, ASPEN (public version). Superstructures of synthesis problems with changes in the topology of the flowsheet can be handled effectively with this new tool. Also special modeling schemes are not needed for splitter and mixers with the proposed implementation. The process synthesis prototype in ASPEN has been illustrated using the example of the HDA process.

Appendix. GBD/OA/ER/AP2 Algorithm and the Decomposition Strategy

Step 1. Set $K = 1$ and $Z^0 = +\infty$. Select an initial flowsheet through the binary variables.

Step 2. Identify the implicit constraints and modify the NLP problem using the IMP-CONS equations.

Step 3. Identify the nonlinear constraints associated with the selected flowsheet NLP problem and the constraints associated with the subsystem NLP problem.

Step 4. Solve the NLP problem for the selected flowsheet to get the value of the objective function Z^k and store the gradient information and the Lagrangian multipliers. If $K = 1$, go to step 5, otherwise to step 7.

Step 5. Pass the information on multipliers to the NLP's for the subsystem with nonexisting units, and solve these NLP's and store the gradient information.

Step 6. Get the **T** matrix using the OA/ER equations for all the nonlinear equations.

Step 7. Check if $Z^K \geq Z^{K-1}$; if true then the optimal solution is Z^{K-1}; else go to step 8. (The above procedure assumes that the successive NLP subproblems are feasible.

If this is not the case, only an integer cut added in the master problem.)

Step 8. Formulate and solve the master problem (GBD/OA/ER/AP2 equations) to obtain the set of new binary variables which defines the new flowsheet structure. Set $K = K + 1$ and return to step 4.

Literature Cited

Beale, E. M. L. Integer Programming. *The State of the Art in Numerical Analysis*; Academic Press: London, 1977; pp 409-448.

Benders, J. F. Partitioning Procedures for Solving Mixed-variables programming problems. *Numer. Math.* **1962**, *4*, 238-252.

Biegler, L. T.; Hughes, R. R. Optimization of Propylene Chlorination Process. *Comput. Chem. Eng.* **1983**, *7*, 645.

Biegler, L. T.; Cuthrell, J. E. Improved Infeasible Path Optimization for Sequential Modular Simulators. *Comput. Chem. Eng.* **1985**, *9*, 257.

Carcotsios, M.; Petrellis, N. C. ACCOPT—An Integrated Software Package for the Optimization of Discrete and Continuous Processes. Presented at the Annual AIChE meeting, San Francisco, 1989.

Douglas, J. M. *Conceptual Design of Chemical Processes*; McGraw-Hill Chemical Engineering Series; McGraw-Hill: New York, 1988.

Duran, M. A.; Grossmann, I. E. An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs. *Math. Program.* **1986**, *36*, 307-339.

Floudas, C. S.; Aggrawal, A.; Ciric, A. R. Global Optimum Search for Nonconvex NLP and MINLP Problems. *Comput. Chem. Eng.* **1989**, *13*, 1117-1132.

Geoffrion, A. M. Generalized Benders Decomposition. *J. Optim. Theory Appl.* **1972**, *10* (4), 237-260.

Grossmann, I. E. MINLP Optimization Strategies and Algorithms for Process Synthesis. *Proceedings FOCAPD'89*; Siirola et al., Eds.; Elsevier: Amsterdam, 1990; p 105.

Gupta, O. K. Branch and Bound Experiments in Nonlinear Integer Programming. Ph.D. Thesis, Purdue University, 1980.

Han, S. P. A Globally Convergent Method for Nonlinear Programming. *J. Optim. Theory Appl.* **1977**, *22*, 297.

Harsh, M. G.; Saderne, P.; Biegler, L. T. A Mixed Integer Flowsheet Optimization for Process Retrofits: The Debottlenecking Problem. *Comput. Chem. Eng.* **1989**, *13*, 947-957.

Kocis, G. R.; Grossmann, I. E. Relaxation Strategy for the Structural Optimization of Process Flowsheets. *Ind. Eng. Chem. Res.* **1987**, *26* (9), 1869-1880.

Kocis, G. R.; Grossmann, I. E. A Modelling/Decomposition Strategy for MINLP Optimization of Process Flowsheets. *Comput. Chem. Eng.* **1989**, *13*, 797-819.

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

Lang, Y. D.; Biegler, L. T. A Unified Algorithm for Flowsheet Optimization. *Comput. Chem. Eng.* **1987**, *11*, 143.

Marsten, R. *User's Manual for ZOOM/XMP*; The Department of Management Information Systems, University of Arizona: Tucson, 1986.

Powell, M. J. D. The Convergence of Variable Matric Methods for Nonlinear Constrained Optimization Calculations. *Nonlinear Programming*; Academic Press: London, 1978.

Schrage, L. *Linear Programming Models with LINDO*; The Scientific Press: 1983.

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 May 2, 1991
Accepted September 27, 1991

SEPARATIONS

Mathematical Analysis on Catalytic Dehydrogenation of Ethylbenzene Using Ceramic Membranes

Jeffrey C. S. Wu* and Paul K. T. Liu
Alcoa Separations Technology, Inc., Warrendale, Pennsylvania 15086

Ceramic membranes capable of performing Knudsen separations were discussed extensively in the literature as a potential candidate for membrane reactor processes. A mathematical model is employed to evaluate the performance of a membrane reactor for the catalytic dehydrogenation of ethylbenzene to styrene. The model previously discussed in the literature has been modified to include side reactions for estimating product selectivities. According to our analysis in a selected case study, an increase ($\geq$5%) in styrene yield over the thermodynamic limit is achieved by a hybrid system, i.e., a fixed bed reactor in conjunction with a membrane reactor. The proposed membrane reactor shows a different behavior in the generation of key side products, i.e., benzene and toluene. The side reaction for toluene is inhibited as a result of the selective removal of hydrogen, while the generation of benzene continues at a reduced rate.

A commercially available ceramic membrane with a multiple-layer, composite, asymmetric structure is selected in this study. It can deliver gas separations according to Knudsen diffusion. Figure 1 shows a schematic of the selected membrane reactor. The membrane tube is packed with granular catalysts. Reactant is fed in the tubular side, while inert purge or vacuum can be applied in the shell side. The permeable membrane can preferentially remove one of the products (e.g., hydrogen in dehydrogenation) and, then, enhance the conversion.

Several mathematical models describing the performance of membrane reactors were developed in the literature (Mohan and Govind, 1986, 1988 a-c; Itoh et al., 1984, 1985; Itoh, 1987, Itoh and Govind, 1989a,b; Sun and Khang, 1988, 1990). Enhanced conversions under various operating conditions were discussed in detail. However, the effect of membrane reactors on product selectivities has not been addressed because none of these models have

0888-5885/92/2631-0322$03.00/0