Carnegie Mellon University
Research Showcase @ CMU

Department of Electrical and Computer Engineering

Carnegie Institute of Technology

1990

# An MINLP process synthesizer for a sequential modular simulator

U. M. Diwekar
*Carnegie Mellon University*

Ignacio E. Grossmann

Edward S. Rubin

Carnegie Mellon University.Engineering Design Research Center.

Follow this and additional works at: http://repository.cmu.edu/ece

This Technical Report is brought to you for free and open access by the Carnegie Institute of Technology at Research Showcase @ CMU. It has been accepted for inclusion in Department of Electrical and Computer Engineering by an authorized administrator of Research Showcase @ CMU. For more information, please contact research-showcase@andrew.cmu.edu.

**NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:**
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

# An MINLP Process Synthesizer for a Sequential Modular Simulator

by

U.M. Diwekar, I.E. Grossmann, E.S. Rubin

EDRC 06-91-90

# An MINLP Process Synthesizer for a Sequential Modular Simulator

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

**Carnegie Mellon University**

**Pittsburgh, PA 15213**

**December 1990**

*Department of Engineering and Public Policy

** Department of Chemical Engineering

**University Libraries**
**Carnegie MeHon University**
**Pittsburgh PA 15213-3890**

Abstract

This paper deals with the development of an MINLP synthesizer for sequential modular simulators. Firstly, a variant of the 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 ASPEN simulator using this algorithm, with the decomposition strategy by Kocis and Grossmann(1989). The application of this new capability is demonstrated with several examples including the structural optimization of the hydrodealkylation of tolune 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 last few years important advances have been made in flowsheet optimization with process simulators using nonlinear programming techniques (NLP). Effective computational strategies based on 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 capability built around the sequential modular simulator ASPEN 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 Vishwanathan and Grossmann (1989) for reducing the effect of nonconvexities in the master problem. Also the extension of GBD using a partitioning 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 (1989) 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 and Biegler (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.

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 (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). This superstructure can then be modeled as an MINLP problem of the form;

$$
\begin{aligned}
& Z = \min_{x,\,y} c^T y + f(x) \\
& \text{subject to} \qquad\qquad \textbf{(MINLP)} \\
& h(x) = 0 \\
& By + g(x) \leq 0 \\
& y \in Y \,;\; x \in X \\
& \text{where } Y = \{y \mid Ay \leq a,\, y \in \{0,1\}^m\};\; X = \{x \mid x^L \leq x \leq x^U,\; x \in R^n\}
\end{aligned}
$$

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 an 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 proceed, 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:**

$$Z^k_{GBD} = \min_{y,\ \alpha_{GBD}} \alpha_{GBD}$$

**subject to** **(GBD)**

$$\alpha_{GBD} \geq c^T y + f(x^k) + \sum_{j=1}^{ti} M_j^k [\, g_j(x^k) - b_j y \,] \qquad k = 1, 2 \ldots K$$

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

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

**where k represents the iteration counter, and $M_j^k$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:**

$$Z^K_{OA} = \min_{y,\ x, \alpha_{OA}} \alpha_{OA}$$

**subject to**

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

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

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

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

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:

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

where $T^k = [\, t^k_{ii} \,]$

$$t^k_{ii} = \begin{cases} 1 \text{ if } \lambda^k_i > 0 \\ 0 \text{ if } \lambda^k_i = 0 \\ -1 \text{ if } \lambda^k_i < 0 \end{cases}$$

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:

$$Z^K_{AP} = \min_{y,\, x, \alpha_{AP},\, p^k,\, q^k,\, s^k} \alpha_{AP} + \sum_{k=1}^{K} (w^k_q)^T q^k + \sum_{k=1}^{K} (w^k_p)^T p^k + \sum_{k=1}^{K} (w^k_s)^T s^k$$

subject to

**(OA/ER/AP)**

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

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

$$B\,y + g\,(x^k) + \nabla g\,(x^k)^T (x - x^k) \leq s^k \qquad k=1,2,\ldots K$$

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

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

where $q^k$, $p^k$, £ are the slack variables with corresponding large finite weights w£, $w^*_{\text{£}}$, $w^k_s$.

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.

### *2.13 GBDIOAIERIAP; 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 flowsheer 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:

$$Z^K_{GOEA} = \min_{y,\, x\, \alpha_{GOEA},\, P} \text{CCGOEA}$$

subject to

$$\text{OGOEA}\wedge cTy + f(x^*) + Vf(_xK)^T(x - X^*) + (w£)T_pK + (w^K_s)^T s^K$$

$$T^K[Vh(x^K)^T(x-x^K)] \leq p^K$$

$$By + g(x^K) + Vg(x^K)^T(x - x^K) \leq, s^K$$

$$\text{OGOEA} * cTy + f(x^k) + Vf(x^k)^T(x-*^k) + \sum_{j=1}^{t1} Hjtbj Y + gj(x^k) + V_{gj}(x^k)^T(x-x^k)] \quad {}_k = 1,2, \ldots K\text{-}l$$

**(GBD/0A/ER/AP1)**

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

**y € Y; x € X; aooEAe $R^1$**

**where Y = {yl Ay ≤ a, y e {0,1 }$^m$ ; X = {xl xL≤ x≤ $x^U$ x e $R^n$}**

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.

$$Z^k Q Q E \wedge \min_{y,\ x,\ CCGOEA,\ P} OQOEA$$

subject to

$$OGOEA \geq C^T y + f(X^1) + Vf(X1)^T(X - xl) + (w_p^1)V + (w_s^1)\ V$$

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

$$B\ y + g\ (x^1) + Vg\ (x^1)^7 \wedge - x^1) \leq s^k$$

$$OCGOEA \wedge cTy + f(x^k) + Vf(x^k)^T(x-x^k) + \sum_{j=1}^{ti} H^h y + gj(^{xk}> + {}^v gj(\wedge^k)^T(^x - {}^{xk})l \qquad k = 2,3,\ldots K$$

**(GBD/OA/ER/AP2)**

$$X \sum_{j e B^k} y \wedge - \sum_{ielSI\wedge} y \wedge <IBH - 1 \qquad u\ \_\ i\ \wedge\ xr$$

$$ye\ \ Y;X€X;OCGOEA€\ \ R^1$$

where $Y = \{y!Ay \leq a,\ y\ \{0,1\ J^*;\ X = \{xlx\wedge x\wedge x^U\ x\ e\ R^n\}$

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 Vishwanathan and Grossmann 1990), this master problem does not necessarily yield non-decreasing 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.

***AnIllustrativeExample:***

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

$$\text{Min} \quad y1 + 1.5y2 + 0.5y3 + x11 + x12$$

subject to

$$\begin{aligned} x11 - x1^2 &= 0 \\ x12 - x2^2 &= 0 \\ x13 - x2 - (x1-2)^2 &= 0 \\ x13 &\geq 0 \\ 2y1 - x1 &\leq 0 \\ 1 - y1 - x1 &\leq 0 \\ 3y3 - x1 - x2 &\leq 0 \\ y1 + y2 + y3 &\geq 0 \\ y1, y2, y3 &= 0,1 \end{aligned} \qquad \text{(EX1)}$$

Table 1 presents the comparison of the step by step results obtained using GBD/OA/ER/AP1, 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 inefficient 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 non-existing units is provided through a Lagrangian

suboptimization procedure which enforces non-zero 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 given below:

$$\text{Min } 7.5y1 + 5.5y2 + 7v1 + 6v2 + 5x$$

subject to

$$\begin{aligned} z1 &= 0.9\,[\,1 - \exp(-0.5\,v1)\,]\,x1 \\ z2 &= 0.8\,[\,1 - \exp(-0.4\,v2)\,]\,x2 \\ x1 + x2 - x &= 0 \\ z1 + z2 &= 10 \\ v1 &\le 10y1 \\ v2 &\le 10y2 \\ x1 &\le 20y1 \\ x2 &\le 20y2 \\ y1 + y2 &= 1 \\ y1, y2 &= 0,1; \quad x1,x2,z1,z2,v1,v2 \ge 0 \end{aligned} \qquad \textbf{(TWO-REA)}$$

The binary variables yl and y2 denote existence of reactors 1 and 2. The material balance constraints for interconnection nodes include splitting of input flows xl and x2, and mixing of outlets zl and z2. This problem was solved using GBD/0A/ER/AP2 algorithm with the decomposition scheme shown in Figure 2(b). 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:

$$\text{Min } 5.5y2 + 6v2+5x$$

subject to

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

$$xl + x2 - x = 0$$

$$zl + z2 = 10$$

$$v2 \wedge 10$$

$$x2 \wedge 20$$

The solution to this NLP is Z = 107.376 at xl= x = 15 and v2 = 4.479 and the Lagrange multipliers for the mixer mass balance (zl + z2 = 10) is -7.5, Linearizations for the nonlinear constraints corresponding to the initial flowsheet results in,

$$z2 - 0.666x2 - 0.800 v2 + 3.584 - p2 \wedge 0$$

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

$$\text{Min} 7.5yl+7vl+5xl-7.5zl$$

subject to

$$zl=0.9[l-\exp(-0.5vl)]xl$$

$$xl-15=0$$

$$vl \wedge 10$$

$$xl \wedge 20$$

Note that the material balance equation for inlet to the interconnection node is replaced by the constraint which fixes the inlet flowrate equal to the inlet flowrate calculated at the optimal value of the initial flowsheet (xl-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 zl = 11.633 and vl = 3.957. The linear approximation to the corresponding constraint results in;

$$zl - 0.776x1 - 0.933vl - 3.693 - pi \wedge 0$$

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 ($y2-yl \leq 0$) is also included. The solution is 95.64 (a-£$^{w}$jft) at yl=l and y2 = 0 indicating the existence of the first reactor and nonexistence of the second one.

HI. 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,

$$a \geq 7.5yl + 5.5y2 + 7vl + 6v2 + 5x - 6.7z2 + 6v2 + 5x2$$

$$yl - y2 < £0$$

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). Appendix I describes the steps involved in this procedure.

# 3. Implementation in ASPEN

## 3.1 Process Synthesis Capability

The MINLP process synthesis capability in ASPEN is based on ZOOM (Marsten 1987), 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.

First 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 non-existing 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 (1989).

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 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 flowsheeet. 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 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 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 and Biegler (1989). This saves on the unnecessary I/O.

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 implicit constraints linearization, 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;

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

subject to

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

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

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

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

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

subject to **(IMP-CONS)**

$$g(u) + b\,y \le 0$$

$$h_i(u) = 0$$

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

$$\mathbf{U \in U,\ V \in V}$$

where h2(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).

### 4.1 Separation Scheme Synthesis:

Consider the problem in Fig. 4 of selecting the optimal separation scheme to be used to separate two components (A and B) which are available in feed streams Fl and F2. The compositions of these streams are 55 % A, 45 % B and 50 % A, 50 % B respectively and the desired product streams are PI and P2. Purity specifications are a minimum of 80 % A in product PI 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

to produce 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 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 2. 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 (1988) 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 of toluene (HDA) 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: *2benzene — > diphenyl + hydrogen.* The conditions for these gas phase reactions are a pressure of 3.45 (Mpa) and a temperature between 1150 to 1300 F. At lower temperatures, the toluene reaction is too slow and at higher temperatures hydrocrackirig 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 kmoles/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 is given in Table 3. The decomposition strategy is used to solve this problem.

The resulting MINLP optimization problem contains 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 3 or less than 3 NLP iterations and 2 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 especial modeling schemes are not needed for splitter and mixers with the proposed implementation.The process synthesis capability in ASPEN has been illustrated using the example of HDA process.

**References:**

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

**Benders, J. F. Partitioning Procedures for Solving Mixed-variables programming problems, *Nwnerische MathematikA** 238-252.**

**Biegler L.T. and J. E. Cuthrell (1985), Improved Infeasible Path Optimization for Sequential Modular Simulators. *Comput. chem. Engng*, 9, 257**

**Biegler L.T. and R.R. Hughes (1983), Optimization of Propylene Chlorination Process. *Comput. chem Engng*, 7, 645.**

**Carcotsios, M. and N. C. Petrellis. (1989). ACCOPT- An Integrated Software Package for the Optimization of Discrete and Continuous Processes. Annual AIChE meeting, San Franscisco,**

**Douglas J. M. (1988). Conceptual Design of Chemical Processes. Me Graw Hill Chemical Engineering Series.**

**Duran, M.A. and I. E. Grossmann. (1986). An Outer-Approximation Algorithm for a Class of Mixed-Integer Nonlinear Programs. *Mathematical Programming*, 36,307-339.**

**Floudas, C.S., A. Aggrawal, and A.R. Ciric (1989). Global Optimum Search for Nonconvex NLP and MINLP Problems. *Comput. chem. Engng.*, 12, 1117-1132**

**Geoffrion, A. M. (1972). Generalized Benders Decomposition. *Journal of Optimization Theory and Applications*, 10(41 237-260.**

**Grossmann I. E. (1990). MINLP Optimization Strategies and Algorithms for Process Synthesis. Proceedings FOCAPD (Eds. Siirola et al.).**

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

Han, S.P. (1977). A Globally Convergent Method for Nonlinear Programming, *Journal of Optmization Theory and Applications*, 22,297\

Harsh M.G., P Saderne, and L. T.Biegler (1989), A Mixed Integer Flowsheet Optimization for Process Retrofits: The Debottlenecking Problem, *Computers and Chemical Engineering*, 12, 947-957,

Kocis, G. R. and I. E. Grossmann (1987). Relaxation Strategy for the Structural Optimization of Process Flowsheets, *Industrial and Engineering Chemistry Research*, 2S£2i» 1869-1880.

Kocis, G. R. and I. E. Grossmann. (1989). A Modelling/Decomposition Strategy for MINLP Optimization of Process Flowsheets, *Computers and Chemical Engineering*, 12,797-819.
Kravanja, Z. and I. E. Grossmann (1989). PROSYN- An MINLP Synthesizer, Annual AIChE meeting, San Franscisco.

Lang, Y. D. and L.T., Biegler (1987). A Unified Algorithm for Flowsheet Optimization. *Comput. chem. Engng.*, JLL, 143

Lang, Y. D. and L.T., Biegler, and I. E. Grossmann (1988). Simultaneous Optimization and Heat Integration with Process Simulators. *Comput. chem. Engng.*, 12, 311-327

Marsten, R. (1986), User's Manual for ZOOM/XMP. The Department of Management Information Systems, University of Arizona.

Murtagh, B.A. and M. A.Saunders (1985), MINOS User's Guide. Technical Report SOL 83-20, Systems Optimization Laboratory, Department of Operations Research, Stanford University.

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

**Schrage L.(1983), Linear Programming Models with LINDO, The Scientific Press.**

**Vishwanathan, J. and I. E. Grossmann. (1989), A Combined Penalty Function and Outer-Approximation Method for MINLP Optimization, to appear in *Computers and Chemical Engineering.***

## Appendix I

### GBP/ OA/ ER/ AP2 algorithm and the Decomposition Strategy

**Step 1 :** Set K=l and Z° = + «>. Select an initial flowsheet through the binary variables.

**Step 2** Identify the Implicit constraints and modify the NLP problem using equation (IMP-CONS).

**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 NLFs for the subsystem with nonexisting units, and solve these NLP's and store the gradient information.

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

**Step 7* :** Check if Z^ $\geq$ Z K-l; if true then the optimal solution is Z K-l; else go to step 7.

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

---

* The above procedure assumes that the sucessive NLP subproblems are feasible. If this is not the case only an integer cut added in the master problem.

**Table 1. Comparison of GBD/OA/ER/AP with OA/ER/AP algorithms (EX1)**

| Major Step | Major Iteration | Objective function | | | Binary variables | | |
|---|---|---|---|---|---|---|---|
| | | OA/ER/AP | GBD/OA/ER/AP1 | GBD/OA/ER/AP2 | OA/ER/AP | GBD/OA/ER/AP1 | GBD/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 |

**Optimal Solution : x1 = 1.0, x2 = 1.0, x11 = 1.0, x12 = 1.0, x13 = 0.0 obj.fun = 3.5**

**Table 2. Results for** the Separation Synthesis Problem

| Initial Point | Iteration | NLP | MILP | Optimum | CPU time* |
|---|---|---|---|---|---|
| YD=0, YF = 0 | 1 | 0 | (0,1) | | 12.62 |
| | 2 | 478 | (1,0) | | |
| | 3 | 482 | **(l.D** | | |
| | 4 | 510 | infeasible | 510 | |
| YD = O, YF=1 | 1 | 478 | (1.0) | | 10.75 |
| | 2 | 482 | (1,1) | | |
| | 3 | 510 | Infeasible | 510 | |
| YD = 1, YF = O | 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 | |

* sees, total-ASPEN run on VAX 3200.

**Table 3. Cost Data for HDA Problem**

| Feedstock or product/byproduct | costs/price | ($/kg-mole) |
|---|---|---|
| Hydrogen feed | 95 % hydrogen<br>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 ($103yr-l) | Fixed-charge cost | Linear coefficient |
|---|---|---|
| Compressor | 7.155 | 0.815 Xbhp(kw) |
| Stabilizing column | 1.126 | 0.375 X number of trays |
| Benzene column | 16.3 | 1.55 X number of trays |
| Toluene column | 3.90 | 1.12 X number of trays |
| Furnace | 6.20 | 1172 X heat duty (106 kJ /yr) |
| Reactor (adiabatic) | 74.3 | 1.257 X reactor volume (m3) |
| Reactor (isothermal) | 92.875 | 1.571 X reactor volume (m3) |

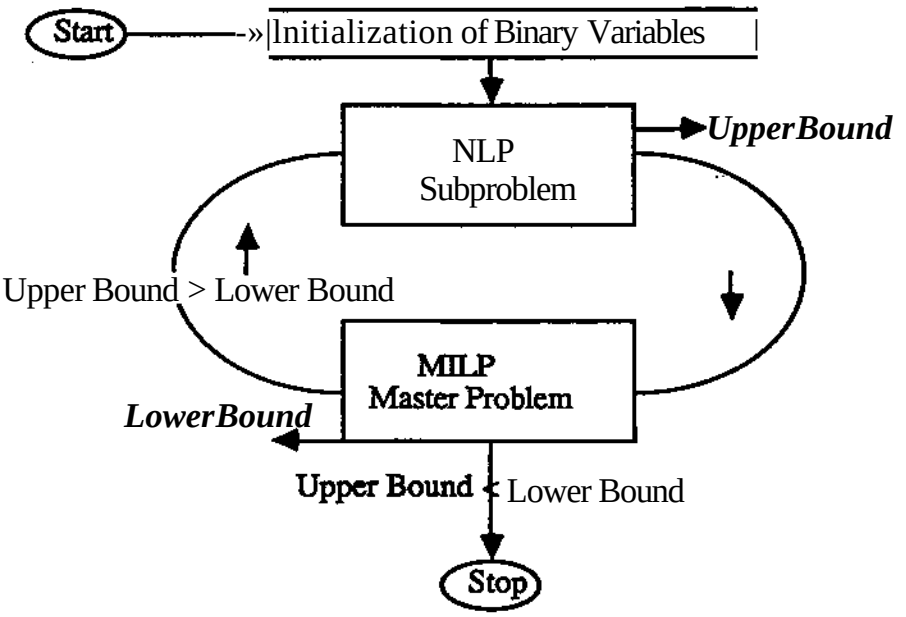
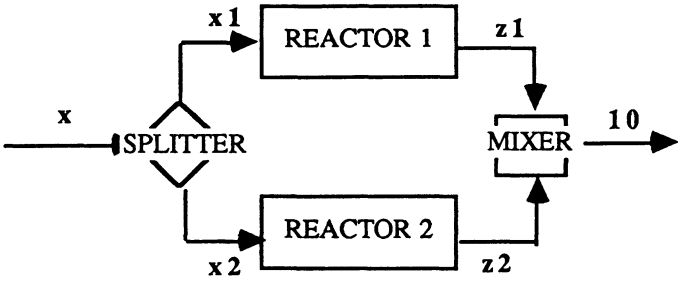
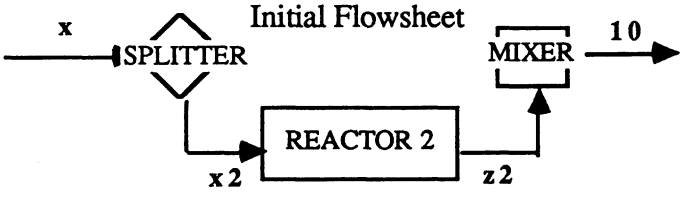
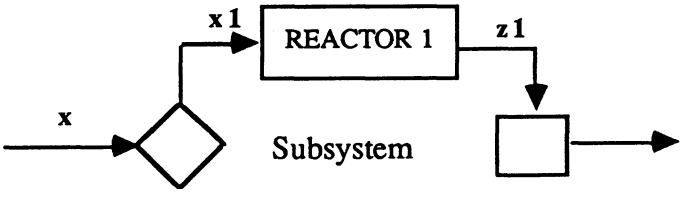



**Figure 1. Main Steps in GBD and OA Algorithms**




**a) Superstructure**




**b) Initial flowsheet and the subsystem**

**Figure 2. Superstructure for the Two Reactor Problem**

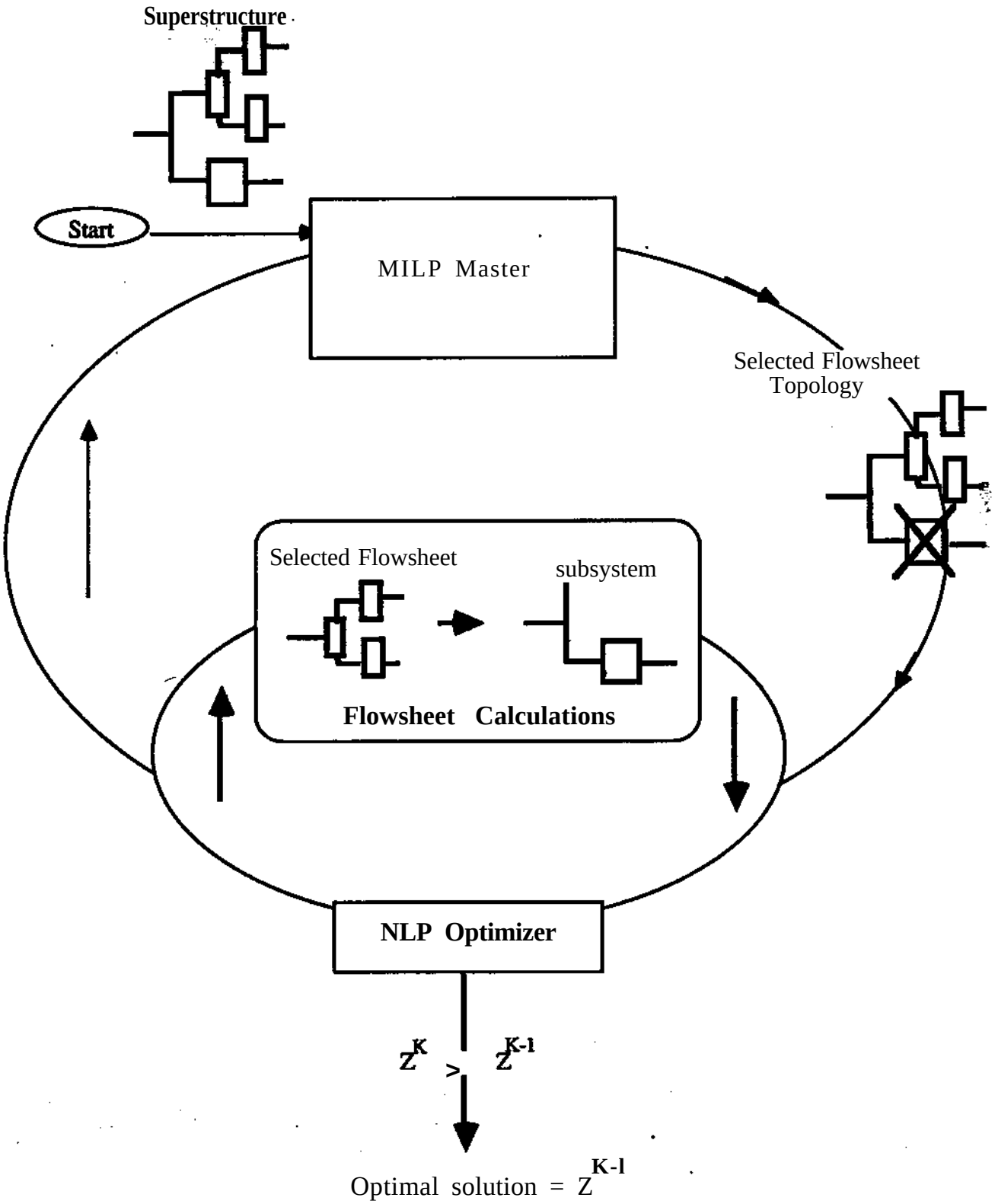



Figure 3. Process Synthesizer for the ASPEN Simulator

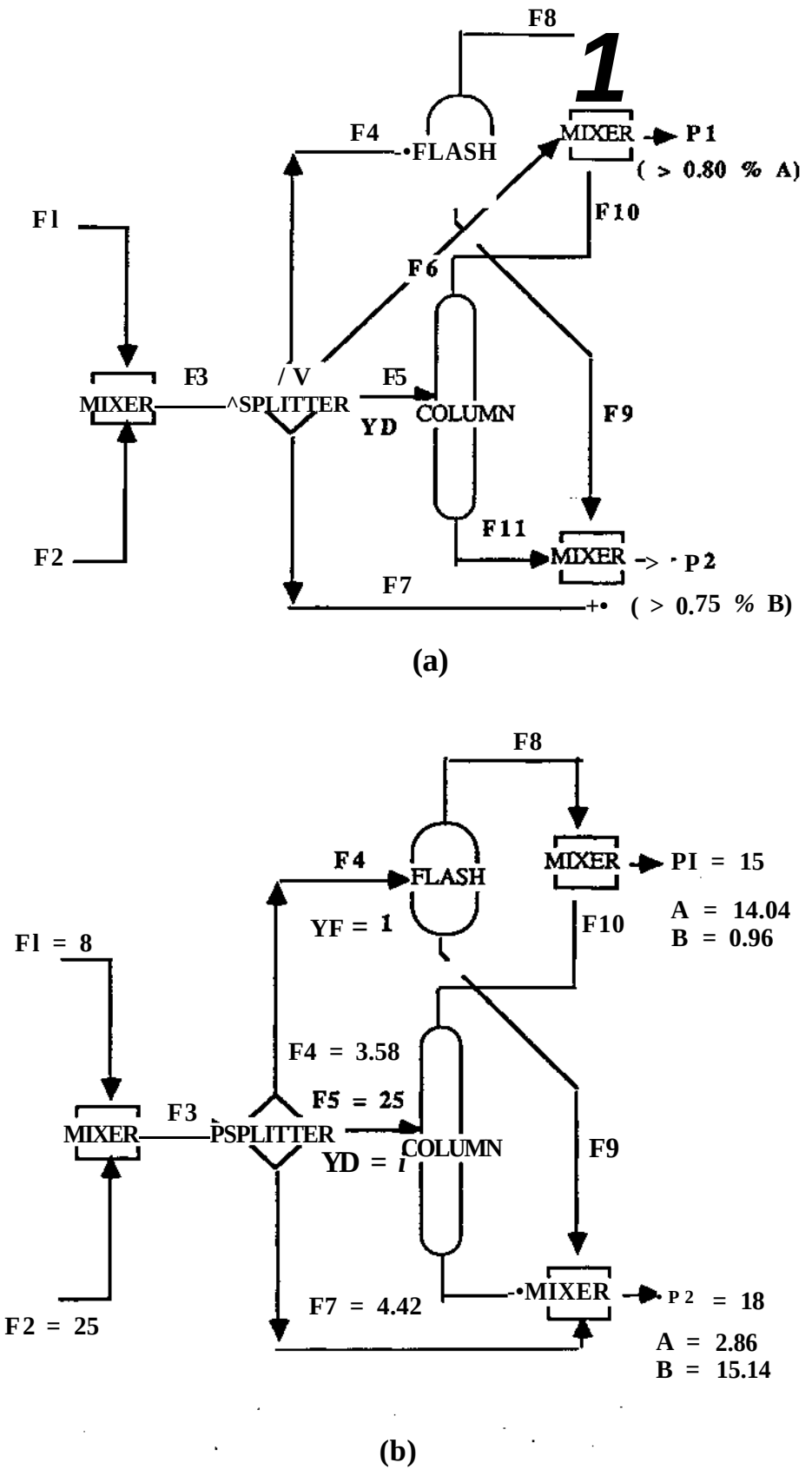



**Figure 4. Superstructure (a) and the Optimal Solution (b) for the Separation Synthesis Problem**

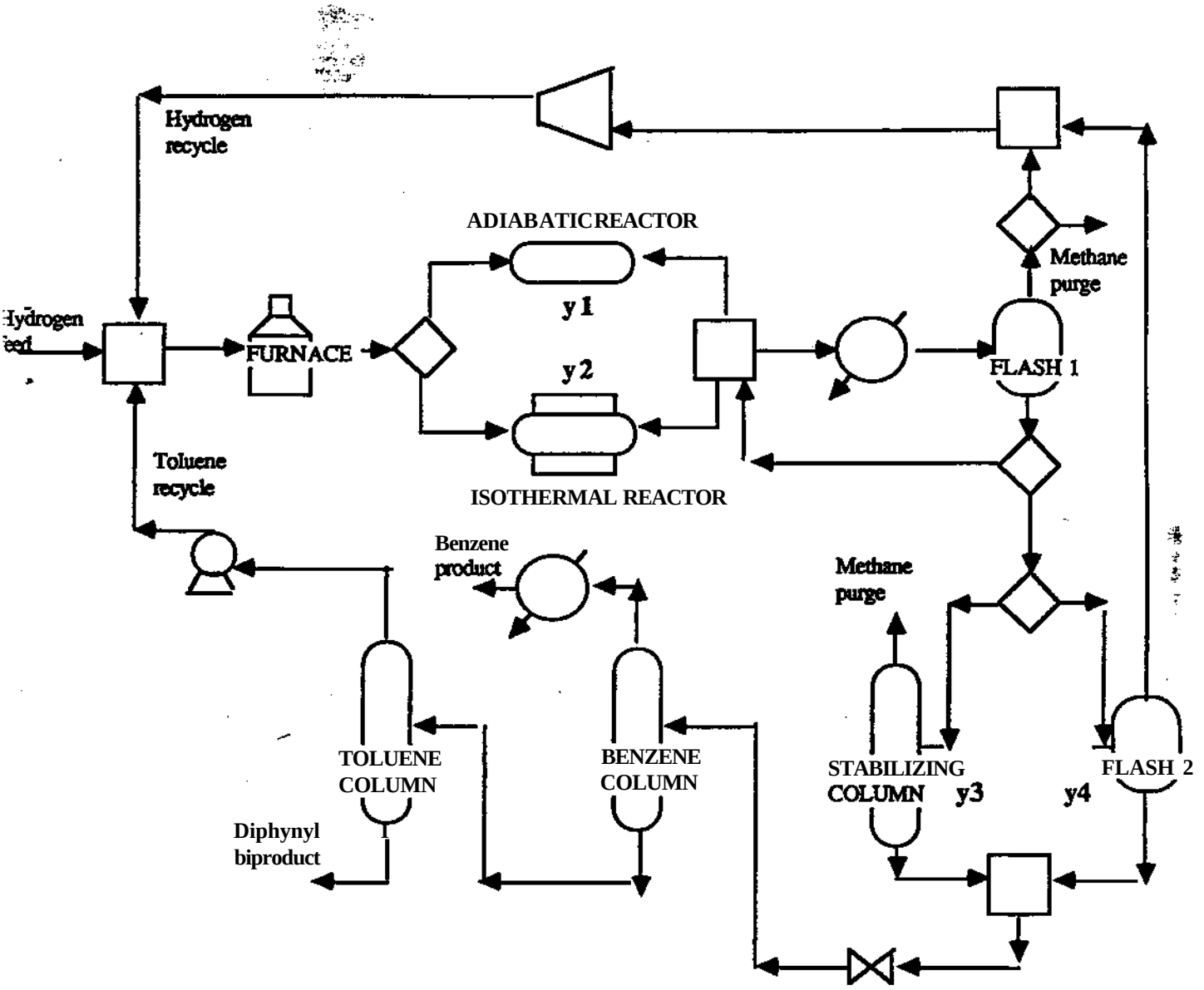



**Figure 5. Superstructure for the HDA Process Synthesis Problem**

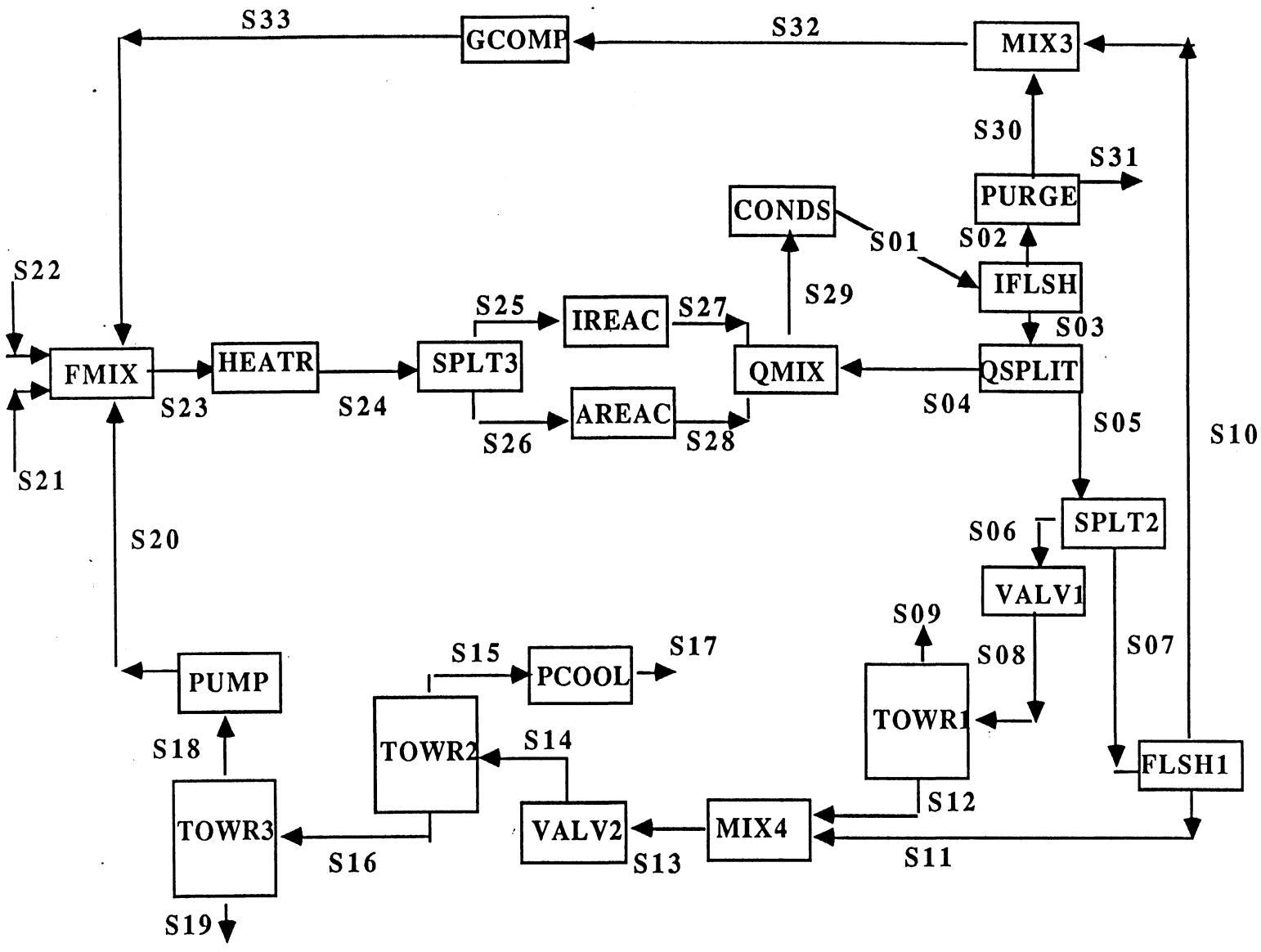



Figure 6. ASPEN Representation of the Superstructure for the HDA Process Synthesis Problem

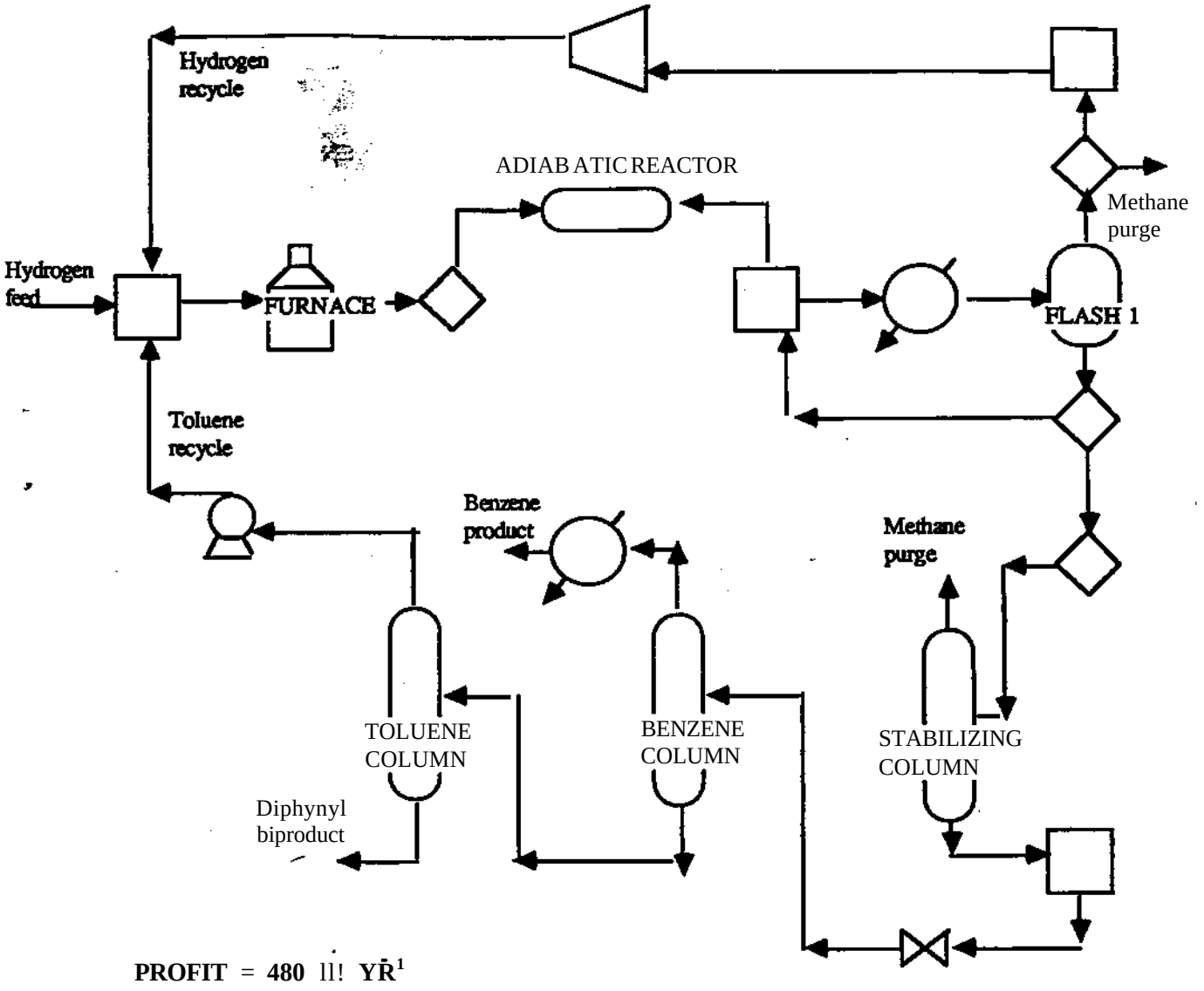



**Figure 7. Optimal Flowsheet Structure for the HDA Process Synthesis Problem for strigent purity requirements ( > 99 % of Benzene)**

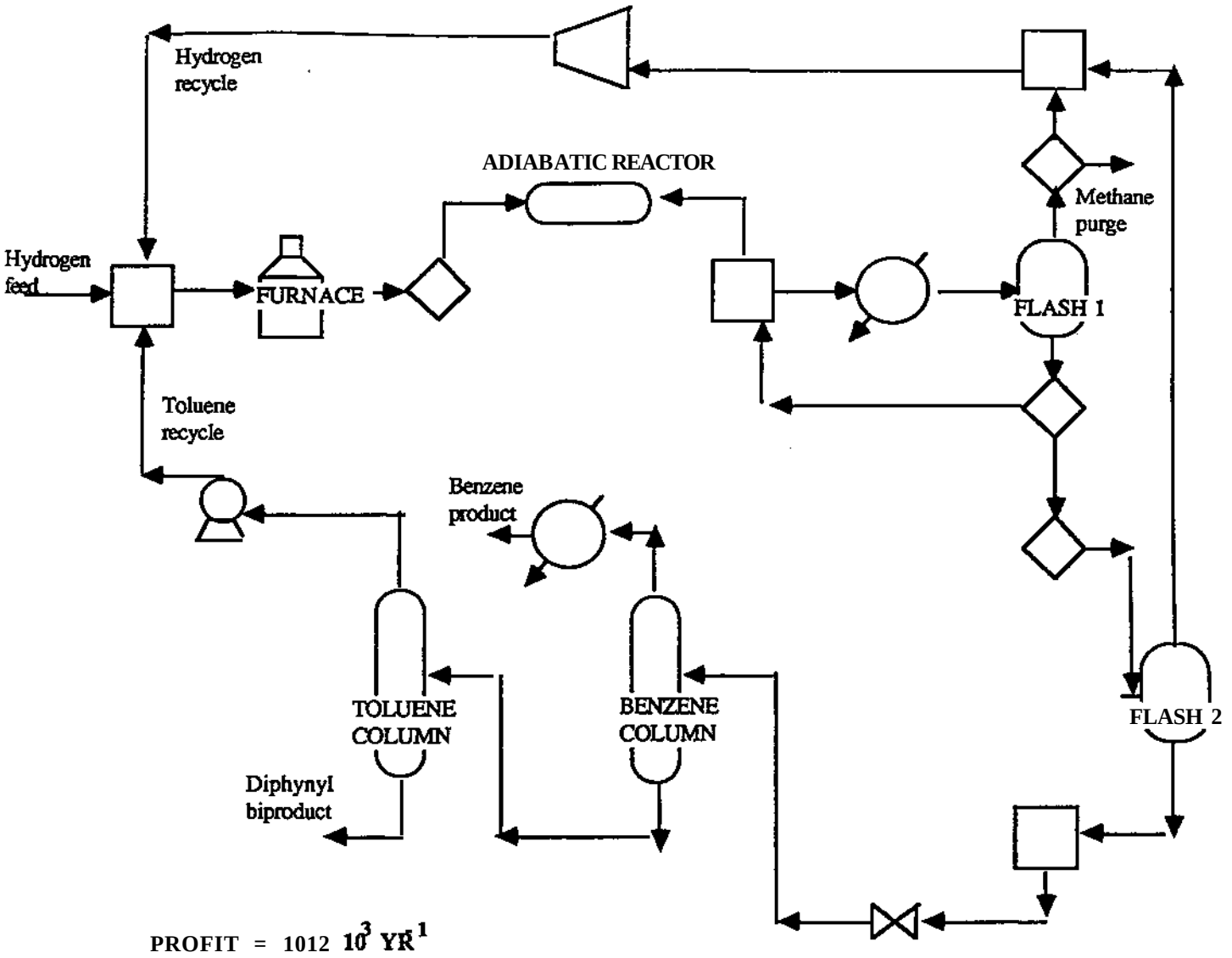



**Figure 8. Optimal Flowsheet Structure for the HDA Process Synthesis Problem for less strigent purity requirements ( > 0.95 % )**