1989

MINLP optimization strategies and algorithms for process synthesis

Ignacio E. Grossmann
Carnegie Mellon University

Carnegie Mellon University.Engineering Design Research Center.

MINLP Optimization Strategies and Algorithms for Process Synthesis

by

Ignacio E. Grossmann

EDRC 06-59-89
Carnegie Mellon University

MINLP OPTIMIZATION STRATEGIES AND ALGORITHMS FOR PROCESS SYNTHESIS

Ignacio E. Grossmann

Department of Chemical Engineering
Carnegie Mellon University
Pittsburgh, PA 15213

Keywords: process synthesis; process integration; design optimization; mixed-integer nonlinear programming; mathematical modelling; design automation.

ABSTRACT

This paper will attempt to show that one of the trends in the area of process synthesis is its gradual evolution towards the mathematical programming approach. This is in part due to the fundamental understanding that has been gained on the nature of the synthesis problems over the last twenty years. However, another major part has to do with the development of new and more powerful mathematical programming algorithms. In particular, the development of new MINLP algorithms, coupled with advances in computers and software, is opening promising possibilities to rigorously model, optimize and automate synthesis problems. A general overview of the MINLP approach and algorithms will be presented in this paper with the aim of gaining a basic understanding of these techniques. Strengths and weaknesses will be discussed, as well as difficulties and challenges that still need to be overcome. In particular, it will be shown how proper problem representations, effective modelling schemes and solution strategies can play a crucial role in the successful application of these techniques. The application of MINLP algorithms in process synthesis will be illustrated with several examples.

INTRODUCTION

After nearly 20 years of research in the area of process synthesis it would seem to be appropriate to ask the questions of how much has been accomplished and how much remains to be done. This paper will not attempt to provide a detailed answer to the first question since that would require an extensive review which is beyond the scope of this paper. Here we will only try to point out some trends that have emerged, and instead try to concentrate on the second question, albeit some personal views on the area. The major premise of this paper is that a next major step in the synthesis area will be geared towards its formalization through the mathematical programming approach, with major driving forces being the increased need for the automation of the synthesis process, and the need to quickly and rigorously explore a larger number of alternatives at the preliminary stages of design in order to improve the economics and other design criteria. We will try to show that new algorithmic developments with MINLP have already started to show their potential for increasing their impact on how synthesis problems might be approached. This however, will not be done at the exclusion of other approaches that

are based on heuristics, AI techniques, thermodynamic targets or hierarchical decomposition, since it will also become clear that there are also some inherent limitations in the mathematical programming approach despite the advances that we can foresee in this area. Thus, there is clearly also the major question of how to properly combine the quantitative and qualitative approaches (see Glover, 1986; Simon, 1987).

This paper will be organized as follows. We will first present a very brief overview of the major trends that have emerged in process synthesis. We will then outline the major steps that are involved in the mathematical programming approach: formulation of a superstructure of alternatives, and modelling and solution of a mixed-integer nonlinear programming (MINLP) problem. We will discuss for the former step, major alternative representations that can be used, and identify the limitations that are present at this point. We will then concentrate on the modelling and on the basic ideas behind several MINLP algorithms that have emerged. The importance of solution strategies will be discussed and illustrated with a number of example problems. Finally, we will summarize this paper by trying to point out some future research directions.

MAJOR TRENDS IN PROCESS SYNTHESIS

Extensive reviews on process synthesis have been reported in a number of papers. For instance, general overviews can be found in Hendry et al. (1973), Stephanopoulos (1981), Nishida *et al* (1981), and Floquet *et al* (1988), while reviews on specific areas can be found in Gundersen and Naess (1988) on heat exchanger networks, and in Westerberg (1985) on separation systems. In addition, many papers have been published in the literature since the general overview papers.

At the risk of oversimplifying the review of previous work, and of not paying due credit to all the original research work, one could try to summarize the status of process synthesis as follows. In terms of area of application, the synthesis of heat exchanger networks is the one that is the best understood and more developed. Here, thermodynamic targets and insights (Hohmann, 1971; Linnhoff and Flower, 1978; Umeda et al, 1979; Linnhoff and Hindmarsh, 1983) have made a great impact, and helped to motivate algorithmic approaches (Cerda and Westerberg, 1983; Papoulias and Grossmann, 1983; Jones and Rippin, 1985; Floudas and Grossmann, 1986; Saboo et al, 1986; Ciric and Floudas, 1989). The extension to more general forms of heat and power integration is also relatively well developed (Nishio et al, 1980; Townsend and Linnhoff, 1983; Papoulias and Grossmann, 1983; Shelton and Grossmann, 1986; Colmenares and Seider, 1987; Swaney, 1988). Separation systems are the next in terms of attention that they have received. Considerable attention has been devoted to some restricted forms of distillation sequences (Hendry Hughes, 1972; Stephanopoulos and Westerberg, 1976; Gomez and Seader, 1976; Andrecovich and Westerberg, 1985; Eliceche and Sargent, 1986; Wehe and Westerberg, 1987; Floudas and Anastasiadis, 1988; Floudas and Paules, 1988; Duran and Flores, 1988) and specialized azeotropic separations (Ryan and Doherty, 1987), where

heuristics, insights and algorithmic methods have emerged. However, there is still significant work to be done in terms of considering complex columns and incorporating separation technologies other than distillation. Reactor networks have received limited attention (Jackson, 1962; Chitra and Govind, 1985; Achenie and Biegler, 1986; Glasser et al, 1987) partly because of the greater difficulty in modelling these systems. Finally, the synthesis of total flowsheets still remains the least well developed area. Here the initial efforts were led by Siirola et al (1971) using an approach based on the general problem solver. This work was followed by Mahalec and Motard (1977) using mainly heuristics and an evolutionary approach. The more recent developments include the hierarchical decomposition scheme by Douglas (1985, 1988) for conceptual process design, and the simultaneous structural and parameter optimization approach by Grossmann (1985) and co-workers. From the above it is clear that the most progress has been made in the synthesis of homogeneous systems, although even in this case the question of how to develop automated synthesis methods to effectively support the decision making of design engineers remains an open question.

In terms of approaches to synthesis there are three major lines of attack that have emerged: (a) the heuristic approach which relies on intuition and engineering knowledge, (b) the physical insight approach which relies on exploiting basic physical principles, (c) the optimization approach which relies on the use of mathematical programming techniques. The advantages and limitations that have been found with each of them are well known and will not be elaborated in this paper (for discussion see for instance Stephanopoulos, 1981; Grossmann, 1985). While physical insights will clearly continue to play an essential role for the understanding of process synthesis problem and motivating novel problem representations, the objective of developing more general and powerful automated synthesis procedures that can systematically examine large numbers of alternatives can only be addressed with a knowledge based approach, or with a mathematical programming approach. Thus, a relevant question to ask at this point is the following: Given advances that can be expected in the near future in Artificial Intelligence and Mathematical Programming, how will these impact the future development of the area of process synthesis? We will leave the conclusive answer to this question to the reader and to the future. We will try to present here the perspective from the Mathematical Programming viewpoint, and only indicate briefly some potential combination with the Al approach.

MATHEMATICAL PROGRAMMING APPROACH

From a conceptual viewpoint, the process synthesis problem can be stated as follows. Given are specifications of inputs that typically correspond to raw materials, and of outputs that correspond to desired products that are to be produced. The problem then consists in integrating a process flowsheet that will convert the inputs into desired outputs so as to meet desired specifications while optimizing a given objective or goal function.

The synthesis of such a system involves the selection of a configuration or topology, as well as its design parameters. That is, one has to determine on the one hand which process units should integrate a flowsheet and how they should be interconnected, and on the other hand one has to determine the sizes and operating conditions of the units. The former clearly imply making discrete decisions, while the latter imply making a choice from among a continuous space. Thus, from a conceptual standpoint the synthesis problem corresponds to a discrete/continuous optimization problem which mathematically gives rise to an MINLP problem.

In general, the major steps involved in approaching this problem are as follows:

Step 1. A superstructure is postulated that has embedded alternatives that are candidates for a feasible and optimal design.

Step 2. The superstructure is modelled as the MINLP problem:

$$
\begin{aligned}
Z = \min\ & C(y, x) \\
\text{s.t. } & h(y,x) = 0 \\
& g(y,x) \leq 0 \\
& y \in \{0,1\}^m \ , \quad x \in \mathbf{R}^n
\end{aligned}
\qquad \text{(MINLP)}
$$

Here y represents a vector of 0-1 variables that denote the potential existence of units (0 not included, 1 included), while x represents a vector of continuous variables which correspond for instance to flows, pressures, temperatures and sizes of equipment. C(y,x) represents the objective function, h(y,x) = 0 the process equations and g(y,x) $\leq$ 0 design specifications and logical constraints. For most of the applications in synthesis the only dominant structure is that the MINLP is most often linear in the 0-1 variables with nonlinearities being present in the continuous variables.

Step 3. The optimal design is extracted from the superstructure by solving the corresponding MINLP problem.

The general approach to process synthesis outlined in the previous steps is not new. Umeda et al. (1972) were probably among the first authors to advocate it. However, at that time the problem was formulated as a nonlinear programming problem involving only continuous variables, and solved with direct search techniques. Next, Papoulias and Grossmann (1983) formulated the synthesis problem as a mixed-integer linear programming problem in order to explicitly handle 0-1 variables, and resorted to standard branch and bound computer codes. It is not until very recently, however, that due to new algorithmic developments (Duran and Grossmann, 1986a,b), the synthesis problem has been explicitly formulated as an MINLP and first successfully demonstrated on a process flowsheet by Kocis and Grossmann (1987).

The two crucial steps in the approach described above are Step 1 for generating the superstructure, and Step 3 for solving the MINLP problem. As it turns out, however, Step 2 is also extremely important because the way one models MINLP problems can have a great impact on the performance of the algorithms. The next section will discuss the question of formulation of superstructures.

SUPERSTRUCTURES

As indicated in the previous section, in order to formulate the synthesis problem as an optimization problem one has to develop a representation containing all the alternative designs that are to be considered as candidates for the optimal solution. Developing an appropriate superstructure is clearly of paramount importance, as the optimal solution that one obtains can only be as good as the representation that is being used. While this point is often regarded as a major weakness of the optimization approach, knowledge based approaches suffer also from the same limitation since these also require a representation of alternatives, which is often implicit.

Superstructure representations can in general be explicit or implicit in nature. The former give in general rise to networks, while the latter give rise to trees. As an example, consider the separation of a single feed of 4 components A,B,C, and D into pure products. As is well known (Hendry and Hughes, 1972), the alternative separation sequences consisting of sharp splitters can be represented through the tree shown in Fig. 1. This tree representation lends itself to decomposition where the discrete alternatives of separation tasks can be enumerated implicitiy through a branch and bound search. However, in using this representation the MINLP problem must be converted into the separable optimization problem over the discrete space Y_{D_i}, i=1..m,

$$
\begin{aligned}
Z = \min_{Y} \sum_{i=1}^{m} & C_i(y_i) \\
\text{s.t.} \sum_{i=1}^{m} & g_{ij}(y_i) \le a_j \quad j=1,\ldots t \qquad (1) \\
& y_i \in Y_{D_i} \quad , \quad i=1\ldots m
\end{aligned}
$$

where continuous variables are selected independently at each node of the tree. Therefore, this formulation is likely to lead to suboptimal solutions even if the branch and bound search is performed rigorously. Note also that in the AI approach, which relies on an implicit enumeration of the discrete alternatives, fewer nodes are examined in the tree with the use of heuristics, which increases further the likelihood of obtaining suboptimal solutions.

On the other hand, consider the network representation shown in Fig. 2 that is implied by the MELP model of Andrecovich and Westerberg (1985). Here in contrast to the tree, every node corresponds to a distinct separator. Furthermore, alternative sequences can be represented by using the same subset of nodes (e.g. Sequence 1: A/BCD, B/CD, C/D, Sequence 2: AB/CD,

A/B, C/D share the node C/D). This network is a more compact representation for modelling the problem explicitly as an MINLP. The advantage here is that the optimization can be performed rigorously as then the continuous variables can be optimized simultaneously with the selection of the configuration. The disadvantage is that one loses the capability of performing the straightforward decomposition that is possible with the tree. It will be shown later in the paper, however, that one can still resort to more sophisticated decomposition schemes to rigorously solve the MINLP problem for the network.

From the above discussion, it then follows that network representations that are explicitly modeled as MINLP problems provide a more general and rigorous framework for the optimization.

The next question to be addressed is how to actually postulate or derive the superstructures. As our experience and the one of other researchers has shown, this is an easier task for homogeneous systems (e.g. heat exchanger networks, distillation sequences, reactor networks) than for heterogeneous systems (e.g. total flowsheets). An example of a homogeneous superstructure is the network in Fig. 2, although it is only restricted to sharp separations. One can however, extend this representation for multiple feeds, with single feed columns that allow the possibility of by-passes and nonsharp splits (e.g. see Floudas and Anastasiadis, 1988).

As an additional example of an homogeneous system, consider the synthesis of heat exchanger networks. Here, one possible representation that allows for stream splitting and mixing is shown in Fig. 3, where each exchanger unit corresponds to a potential match of pairs of streams (see Floudas *et al*, 1986, Ciric and Floudas, 1988b). A more general and richer representation for the same problem where the layout and pipe connections can be accounted for, is shown in Fig. 4 where one only has to specify the maximum number of units in the network (see Yee and Grossmann, 1988). Note that in this case aside from all the alternatives of Fig. 3, there is the possibility of rematching the same pair over several exchangers, and to even mix different process streams or represent multi-stream heat exchangers (see Yee and Grossmann, 1989).

It is interesting to note in the previous example, that in the superstructure of Fig. 3 there is a one-to-one correspondence between the units and the tasks they can perform (e.g. unit 1 involves a match of H1 and C1). On the other hand in the superstructure of Fig. 4 there is a one-to-many relationship between the units and the tasks (e.g. unit 1 can perform matches H1-C1 or H1-C2). Another example of a one-to-many relationship, is the superstructure for separation in Fig. 5 proposed by Sargent and Gaminibandara (1976), which in addition to accomodating sharp and non-sharp splits has embedded Petlyuk columns as alternative designs. From these examples it is clear that superstructures that have one-to-many relationships between units and tasks tend to be richer in terms of alternatives that they have embedded. On the other hand, the more restricted one-to-one superstructures tend to require simpler MINLP models that are quicker to solve.

Also, it should be noted that higher level representations of these superstructures can be developed which aggregate the components and streams of these detailed superstructures. Here again the example per excellence is the heat exchanger network problem, where the targets for minimum utility cost and minimum number of units can be modelled respectively as LP and MDLP transportation (Cerda and Westerberg, 1983) or transshipment problems (Papoulias and Grossmann, 1983). For the case when the stream data are variable (flowrates and temperatures), the utility target problem can be modelled as a system of inequalities (see Duran and Grossmann, 1986c), while for the case when only the flowrates are variables it can be represented by the transportation or transshipment equations (see Papoulias and Grossmann, 1983; Andrecovich and Westerberg, 1985). While these representations clearly simplify the synthesis problem, they also have the drawback that they do not provide all the explicit information to synthesize a network.

To systematically develop superstructures for heterogeneous systems is in principle a more difficult task. For instance consider a process flowsheet that is composed of reaction, separation and heat integration subsystems. One could in principle develop a superstructure by combining the superstructures for each subsystem. This, however, could lead to a very large MINLP optimization problem.

Therefore, the major approach that has been taken up to know is to assume that some preliminary screening is performed (e.g. through heuristics) in order to postulate a smaller number of alternatives in the superstructure (Kocis and Grossmann, 1987). While this approach would seem to be restrictive, it does provide a systematic framework for analyzing specific alternatives at the level of tasks. As an example, consider the synthesis of a plant where a preliminary screening would indicate that the major options are as follows: single or two stage compression for the feed, three possible reactor types, possible use of membrane separator for the purge stream, use of flash separation with the option of absorption/distillation columns. Figure 6 displays the superstructure for these altematives. This superstructure has actually embedded a minimum of 24 different configurations. As another example, Fig. 10a shows a superstructure for the HDA process developed by Kocis and Grossmann (1988b) based on the alternatives that were postulated by Douglas (1988) in the hierarchical decomposition scheme. Thus, generating superstructures for heterogeneous systems based on specific alternatives at the level of tasks is actually not a very difficult problem. However, it is clear that in order to consider a larger number of alternatives, the more natural approach would be to combine all the superstructures of the homogeneous subsystems. At this point, it is an open question as to whether this will require the capability of solving much larger MINLP problems, the development of strategies to successively aggregate and disaggregate subsystems, or else a combination with AI techniques (Beltramini *et al,* 1989) or with the hierarchical decomposition scheme by Douglas (1985) to systematically eliminate alternatives from a superstructure that potentially has a very large number of alternatives.

MINLP MODELLING OF SUPERSTRUCTURES

Having developed a superstructure for the candidate designs to be considered, the next step involves the modelling of the MINLP optimization problem. The major feature in such models is the modelling of discrete decisions which is typically performed with 0-1 variables. For most applications it suffices to assign these variables to each potential unit in the superstructure as the interconnecting streams are activated or deactivated according to the selection of units. There are, however, cases when it is also necessary to assign 0-1 variables to the streams (e.g. see Yee and Grossmann, 1989).

The handling of 0-1 variables allows the specification of constraints which are extremely relevant for synthesizing practical flowsheet structures. Typical examples include the following:

a) Multiple choice constraints for selecting among a subset of units I:

Select only one unit: $\sum_{i \in I} z^{\wedge} y f^{1} \wedge$ 0)

Select at most one unit: $\sum J, - \wedge 1$ (3)

Select at least one unit: $\wedge 7 / \wedge 1$ (4)

b) If then conditions:

If unit k is selected then unit i must be selected:

$$Xlc-Xi^{\wedge}O \qquad (5)$$

In addition, 0-1 variables can be related to activate or deactivate continuous variables, inequalities or equations. As an example consider the condition for the continuous variable x:

$$\text{If} \quad y = 1 \rightarrow L \leq x \leq U, \quad \textbf{if } y = 0 \rightarrow x = 0$$

which can be modelled through the constraint

$$Ly \leq x \leq Uy \qquad (6)$$

The above constraint is often used in conjunction with cost models with fixed cost charges which again requires the use of 0-1 variables (see Garfinkel and Nemhauser 1972, Nemhauser and Wolsey, 1988). Furthermore, it has been recently shown by a number of authors (e.g. Cavalier and Soyster, 1987) that virtually any prepositional logic statement can be systematically translated into a set of linear inequalities involving 0-1 variables.

While the use of 0-1 variables introduces a very important capability in the modelling of MINLP problems, it is also true that the way one models an MINLP can have a great impact on the performance of the MINLP algorithm. This phenomenon has been widely recognized in the field of integer programming (e.g. see Williams, 1978; Nemhauser and Wolsey, 1988). As an example consider the logical constraint that may arise in a multiperiod problem:

$$\sum_{i=1}^{m} y_i - mz \leq 0 \qquad (7)$$

where z represents the selection of a given unit and yj the operation of the unit in periods i, i=l,...m. This constraint simply states that if z=0 no operation of the unit is possible in the m time periods, while if z=l operation in any of the m periods is possible. While (7) is a "legitimate" constraint, it turns out that its equivalent representation by the set of inequalities

$$y_i z \leq 0 \qquad i=1,\ldots m \qquad (8)$$

is a much more effective way to model this constraint since its relaxation with continuous variables y_i introduces a greater number of extreme points with 0-1 values (see Fig. 7) which greatly reduces the branch and bound enumeration procedure. Another typical example in modelling is the use of a tight upper bound U in the logical constraint, $x - Uy \leq 0$, to tighten the relaxation problem where the 0-1 variables are treated as continuous.

Some of these empirical observations have led to the theoretical study in integer linear programming of facets of 0-1 polytopes that define the convex hull of integer programming problems (Schrijver, 1986). Algorithms are starting to emerge which can systematically generate approximations to these type of constraints, and hence reformulate a "badly" posed integer programming problem in order to tighten the continuous relaxation (e.g. see Crowder *et al,* 1983, for unstructured 0-1 linear problems, Van Roy and Wolsey, 1987, for MILP problems).

In MINLP, however, there is the additional complication that nonlinearities can also be often formulated in many different ways which are equivalent, and as expected this can also have a great impact on the performance of MINLP algorithms. In general, three major empirical guidelines for a "good" MINLP formulation are the following:

1. Try to keep the problem as linear as possible.
2. Try to develop a formulation that has as tight an NLP relaxation as possible.
3. If possible, reformulate the MINLP as a convex programming problem.

The motivation behind these guidelines requires some basic understanding of the MINLP algorithms which we will cover in the next section.

MINLP ALGORITHMS

BASIC ALGORITHMS

While there is a vast body of literature on LP, NLP, and on integer LP with special

MINLP is regarded as a very difficult problem since it corresponds to an NP-hard problem (Nemhauser and Wolsey, 1988) that is prone to combinatorial explosion for large problems. In our view, however, it is a mistake to regard these problems as "unsolvable". Not only are the applications for MINLP extremely rich, but with current methods and technology one can in fact already solve problems of significant size and complexity as will be shown later in the paper. Futhermore, with advances in new algorithms and computer architectures it is reasonable to assume that over the next decade we will see increases in the order of magnitude of sizes of problems that can be currently solved (e.g. see Pekny and Miller, 1989).

Our objective in this section will be to provide a general overview of the basic MINLP algorithms, and emphasize their basic ideas and properties. Firstly, for convenience in the presentation we will assume that the MINLP has the restricted form where the 0-1 variables appear in linear form and where no equations are involved:

$$
\begin{aligned}
Z = \min\ & c^T y + f(x) \\
\text{s.t. } & By + g(x) \leq 0 \qquad\qquad (PI) \\
& y \in Y \quad , \quad x \in X
\end{aligned}
$$

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

Major algorithms for solving the MINLP problem in (PI) include the following:

a) Branch and bound (Beale, 1977; Gupta, 1980)

b) Generalized Benders Decomposition (Benders, 1962; Geoffrion, 1972)

c) Outer-Approximation (Duran and Grossmann, 1986b)

The branch and bound method for MINLP is a direct extension of the linear case (MILP). This method starts by relaxing the integrality requirements of the 0-1 variables which leads to a continuous NLP optimization problem. It then continues by performing a tree enumeration where a subset of 0-1 variables are succesively fixed at each node. The solution of the corresponding NLP at each node provides a lower bound for the optimal MINLP objective function value. This lower bound can then be used to expand the nodes in a breadth first enumeration (i.e. expand the node with lowest lower bound), or else to fathom nodes in a depth first enumeration whenever the lower bound exceeds the best current upper bound. Clearly the size of the tree is dependent of the number of 0-1 variables (maximum 2^{m+}*), although of course the objective in the search is to hopefully enumerate only a small subset of nodes.

The major disadvantage of the branch and bound method is that it may require the solution of a relatively large number of NLP subproblems which cannot be updated as readily as in the linear case where few pivot operations are required to update the LP solution of a new node. On the other hand, if the MINLP has a tight NLP relaxation the number of nodes to be

enumerated may be modest. In the limiting case where the NLP relaxation exhibits 0-1 solutions for the binary variables (convex hull formulation) only one single NLP problem need to be solved.

In contrast to the branch and bound method, both the Generalized Benders Decomposition (GBD) and Outer-Approximation (OA) algorithm consist of solving at each major iteration an NLP subproblem (with all 0-1 variables fixed) and an MILP master problem as shown in Fig. 8. The NLP subproblems have the role of optimizing the continuous variables and provide an upper bound to the optimal MINLP solution. The MILP master problems have 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 the GBD and OA method lies in the definition of the MILP master problem. In the case of GBD it is given by a dual representation of the continuous space, while in the case of the OA it is given by a primal approximation. In particular, given solutions x^k with multipliers $|i^k$ of the NLP subproblems for fixed y^k, k=l,...K,

$$Z(y^k) = \min_x c^T y^k + f(x)$$

$$\text{s.t. } g(x) \leq -By^k \qquad (9)$$

$$x \in X$$

the master problem of GBD is given by

$$z\pounds_B = \min_{GB} o c_{GB} \qquad (10)$$

$$\text{s.t. } a_{GB} \pounds c^T y + f(x^k) +]\underline{T}\, \wedge^k (_{gj}(x^k) - b_{jy}) \quad k=l..K$$

$$y \in Y \quad , \quad a_{GB} \in R^1$$

while the master problem of OA is given by

$$Z_{OA}^{K} = \min_{y,x,\alpha_{OA}} \alpha_{OA} \tag{11}$$

$$\left.\begin{array}{l} \text{s.t. } c^T y + f(x^k) + \nabla f(x^k)^T (x - x^k) - \alpha_{OA} \leq 0 \\ \\ By + g(x^k) + \nabla g(x^k)^T (x - x^k) \leq 0 \end{array}\right\} \quad k=1..K$$

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

Note that in both methods the MILP master problems in (10) and (11) accumulate new constraints as iterations proceed. GBD accumulates one Lagrangian cut in the space of the 0-1 variables, while OA accumulates a set of linear approximations of the nonlinear constraints in the space of both the 0-1 and continuous variables. It can be actually proved (see Duran and Grossmann, 1986b), that each Lagrangian cut in GBD represents a surrogate constraint of the corresponding linear approximations in OA. Therefore, since the master problem of the OA method is richer in information, it can be proved that it predicts stronger lower bounds than GBD and therefore it requires fewer major iterations (Duran and Grossmann, 1986b), This then implies that the OA method requires the solution of fewer NLP subproblems, which in addition become successively easier to solve as the MILP master problem also predicts values of continuous variables which provide excellent initial guesses (Kocis and Grossmann, 1989). On the other hand, it is also clear that the computational demands on the MILP master problem of OA are greater since when compared to the master of GBD, it contains the continuous variables as well as a larger number of constraints. Furthermore, it is clear that the advantage in GBD is that its master problem contains only the 0-1 variables and one scalar variable as well as fewer constraints.

In terms of convergence, neither GBD nor OA have the property that the convex hull formulation converges in one single major iteration as would be the case in the branch and bound method. Here instead the OA algorithm converges in one major iteration if the MINLP reduces to an MILP as then the master problem in (11) provides an exact representation. In the case of GBD, convergence cannot be guaranteed in one major iteration for this limiting case. However, as has been shown by Magnanti and Wong (1981), the optimal formulation for GBD in terms of number of major iterations is the convex hull formulation for which in practice convergence is often achieved in few iterations.

As for sufficient conditions for convergence to the global optimum, all the above algorithms require that the functions in (MINLP) satisfy some form of convexity conditions. The specific requirements vary with each algorithm. For instance, since the OA algorithm is based on the construction of supporting hyperplanes with function linearizations, strict convexity is required by the functions that involve the continuous variables. On the other hand, the branch and bound method requires that each of the NLP subproblems have a unique solution, and therefore strict convexity is not required. Finally, in the case of GBD strict convexity is required

for fixed values of the binary variables, and strict convexity for the Lagrangian function in terms of the 0-1 variables (Geoffirion, 1972). Note, that this condition is not as stringent as for the OA algorithm.

EXTENSIONS

The three MINLP algorithms described above can be extended to explicitly handle nonlinear equations h(x) = 0. In the case of branch and bound this is simply accomplished by appending these equations to the relaxed NLP subproblems that are solved at each node. For the case of GBD no modification is required in the master problem (10), as the multipliers |i of the NLP subproblem with the equations will reflect their effect. In the case of the OA algorithm, handling of equations in the master problem can be accomplished with the equality relaxation strategy by Kocis and Grossmann (1987) (OA/ER algorithm). Here, linearizations of the equations at the solution of the NLP subproblem k, are added to the master problem in (11) by relaxing them according to the sign of the Lagrange multipliers; that is,

$$T^k \left[\nabla h(x^k)^T (x - x^k) \right] \leq 0 \qquad (12)$$

where $T^k = [t \, \pounds \, 1]$ and

$$t_{ii}^k = \begin{cases} 1 & \text{if } \wedge^k > 0 \\ -1 & \text{if } \wedge^k < 0 \\ \mathbf{0} & \mathbf{if \ \ A.f = 0} \end{cases}$$

As has been shown by Kocis and Grossmann (1987) sufficient conditions for global optimality with the OA/ER algorithm require quasi-convexity of the relaxed nonlinear equations.

In addition to the above cited algorithms, a number of extensions have been suggested recently. In the case of GBD, Floudas *et al* (1988) have proposed strategies to partition the continuous variables into complicating variables that are introduced in the master problem, and noncomplicating variables that are optimized in the NLP subproblem. In this way, MINLP problems that are linear in the 0-1 variables and bilinear in the continous variables can be decomposed into continuous LP subproblems (fixed 0-1 and complicating continuous variables) and MDLP master problems (involving the 0-1 and complicating continuous variables). Global optima can then be obtained for each, the LP subproblem and the MILP master, respectively. However, theoretically this does not imply that the global optimum of the MINLP can be attained since the lower bound from the master might not always be valid due to nonconvexities that are introduced in the Lagrangian function. Nevertheless, computational results reported by Ciric and Roudas (1988a) and Floudas *et al* (1988) seem to indicate that the success ratio is very high which is most probably due to the loose approximations in the master problem with which a larger number of integer points is examined.

As for the case of the OA/ER algorithm, Kocis and Grossmann (1988a) proposed a two phase strategy in which nonconvexities are identified numerically with local and global tests in the first phase where the OA/ER algorithm is applied. In the second phase, linearizations of the constraints that are identified as being nonconvex, are relaxed with slack variables which are introduced with a penalty function in the master problem. Also, at this stage, since the master problem is not guaranteed to predict rigorous lower bounds, the termination criterion is changed to one where the cycle of major iterations continues until the NLP subproblem fails to decrease the objective function value. This strategy was shown to yield the global optimum in 80% of a set of test problems.

Recently, Viswanathan and Grossmann (1989) have developed a new variant of the OA/ER algorithm that makes use of an augmented penalty function in the master problem (AP/OA/ER algorithm). The algorithm does not require that an initial guess of the 0-1 variables be supplied as it starts by solving the relaxed NLP problem. If an integer solution is found the algorithm stops. Otherwise it proceeds to formulate an MILP master problem that is similar in nature to the phase two master by Kocis and Grossmann (1988a). However, instead of trying to identify nonconvex linearizations, slacks are added to all the linearizations of the nonlinear relaxed equations and inequalities, yielding the master problem:

$$
\begin{aligned}
& Z^k_{AP} = \min_{y,x} \alpha_{OA} + \sum_{k=1}^{K} \left[w^k_q q^k + (w^k_p)^T p^k + (w^k_s)^T s^k \right] \\
& \text{s.t.} \quad \left.\begin{aligned}
& c^T y + f(x^k) + V/(x^k)^T (x - x^k) - a_{0A} \le q^k \\
& T^k \, p7h(x^k)^T (x - x^k)l_J \le p^k \\
& By + g(x^k) + Vg(x^k)^T (x - x^k) <; \; s^k
\end{aligned}\right\} \quad k=1,..K \qquad (13) \\
& y \in Y, \quad x \in X, \quad a_{0A} \in R^1 \\
& q^k, \; p^k, \; s^k \ge 0, \qquad k=1..K
\end{aligned}
$$

where q^k, p^k, s^k are slack variables with corresponding large finite weights wJ,W^{k}p,w^

For the general case, the cycle of major iterations in AP/OA/ER proceeds until there is no decrease in the NLP solution. It is interesting to note, however, that if the convexity conditions are satisfied, the MILP master problem predicts rigorous lower bounds and in this case this algorithm reduces to the OA/ER algorithm, except that it uses as a starting point the solution of the relaxed NLP. Computational experience with this method has shown a high degree of robustness with nonconvex problems.

Among other important extensions, Yuan *et al* (1987) have extended the OA algorithm for the case when the 0-1 variables are nonlinear. This is simply accomplished by linearizing the 0-1 variables in the master problem. Convergence to the global optimum can be guaranteed for the case when these functions are strictly convex. For the branch and bound method, Ostrovsky

et al (1989) have proposed a strategy in which, at each node corresponding to the best bound, the relaxed NLP solution is rounded in order to compute an upper bound. The search is terminated when the lower and upper bounds lie within a specified tolerance. This strategy, however, would seem to be only suitable for the case when there is a small gap in the relaxed NLP solution. Finally, Mawengkang and Murtagh (1986) and Mawengkang (1988) have proposed a feasibility technique where the main idea is to round the relaxed NLP solution to an integer solution with the least local degradation. This is accomplished through the computer code MINOS by successively forcing superbasic variables to become nonbasic based on information of the reduced costs. While this method has no guarantee of global optimality, it has shown to have very good performance on a set of test problems with modest computational effort (typically 50% of time over the relaxed NLP problem).

COMPUTATIONAL EXPERIENCE

While until very recently there was very little experience reported in the literature for solving MINLP problems, this situation has undergone a significant change over the last few years with developments in algorithms for MINLP and computer software for NLP (Murtagh and Saunders, 1985; Han, 1976; Powell, 1976; Vasantharajan *et al* 1989), MILP (MPSX, ZOOM), and modelling systems such as GAMS (Brooke *et al,* 1988) which have facilitated the implementation of MINLP algorithms (e.g. Paules and Floudas, 1989; Kocis and Grossmann, 1989).

For instance, Table 1 presents computational results with DICOPT++ (Viswanathan and Grossmann, 1989) where the AP/OA/ER method has been implemented as part of GAMS using MINOS for the NLP optimization and MPSX for the MILP master problems. The 18 test problems in Table 1 involve a variety of applications: e.g. planning, flexibility analysis, retrofits of heat exchanger networks, batch design problems, complex column designs and flowsheet synthesis problems. As can be seen, problems with up to 60 0-1 variables and 700 constraints and variables require reasonable computational effort. Additional computational experience with MINLP can also be found in Kocis and Grossmann (1989) and Sahinidis and Grossmann (1989). The latter authors have reported the solution of an MINLP scheduling model for continuous parallel production lines involving 780 0-1 variables, 23,000 continuous variables and 3,200 constraints. GBD was used by exploiting the structure of the NLP subproblem whose dual solution can be obtained very efficiently.

Among the major trends that can be identified from the experience in solving MINLP problems, we can cite the following.

Firstly, problem formulation is one of the most crucial aspects for the successful solution of MINLP problems. Major features that can greatly enhance the efficient solution and convergence to the global optimum are tight NLP relaxations which can be accomplished for instance by tightly bounding the continuous variables, specifying smallest upper bounds in logical constraints such as in (6) and replacing weak integer constraints such as in (7) by a

stronger set of inequalities. This has the effect of reducing the number of branches or major iterations in GBD or in the OA variants. To maximize the occurrence of linear constraints and minimize the occurrence of nonlinear functions is another desirable guideline for modelling, as this enhances the robustness of the solution of the NLP subproblems and tends to minimize the effect of nonconvexities. Also, one should avoid if possible the use of products of 0-1 variables with continuous variables or functions as this often introduces nonconvexities. Lastly, one should try to reduce the number of 0-1 variables by exploiting the connectivity in a superstructure; this has the obvious effect of reducing the potential size of the tree that is to be examined by the branch and bound methods for MINLP and MILP.

Secondly, no algorithm is consistently superior in all the applications. Branch and bound is clearly superior if the relaxed NLP happens to exhibit integer solutions. As this is usually not the case, both GBD and the variants of the OA algorithm will normally outperform branch and bound which in large problems may require the solution of hundreds or thousands of NLP subproblems. As for the two latter algorithms, the family of OA algorithms will normally require much fewer major iterations (typically 3 to 5) than GBD, although the expense in solving the MILP master problem will be greater. For modest number of binary variables this is often not a serious limitation. However, this can become the major bottleneck in the computations if the relaxation in the MILP master is poor and the number of integer variables is large.

A distinct advantage with GBD, is that special structures can be exploited more readily. For instance, fixing a subset of complicating variables the resulting subproblem might reduce to an LP, to a convex NLP or to a set of disjoint problems that can be solved in parallel (Geoffrion, 1972). However, although the advantage in GBD is that the MILP master problem is easier to solve than in the OA methods, the number of major iterations with GBD can be somewhat unpredictable (many cases 5 to 20, but sometimes up to one hundred with many infeasible NLP subproblems). This can be a serious limitation if the NLP subproblems are expensive to solve. One way to reduce the number of major iterations in GBD is to either add extra constraints to the master problem and/or define some of the continuous variables as complicating variables for the master problem (e.g. see Yee and Grossmann, 1988; Sahinidis *et al*, 1989; Ciric and Floudas, 1988b) which then however increases the expense in solving the MILP.

Thirdly, nonconvexities can cause difficulties in two ways. Firstly, the solution to the NLP subproblems may not be unique, and secondly the master problems in GBD and in the OA variants may cut-off the optimal solution. At present, a promising method to overcome nonconvexities for bilinear NLP subproblems is the partitioning scheme for GBD by Floudas *et al.* (1988), who have demonstrated the effectiveness of this scheme for the NLP optimization of the superstructure for heat exchanger networks (Ciric and Floudas, 1988a). As for the master problem, due to the strong convexity assumptions, the original OA and OA/ER algorithms are the most sensitive to nonconvexities, while GBD tends to be the least affected by this problem. This is due to the fact that convexity is only required in the Lagrangian function, and that the master problem in GBD is poorly constrained and therefore has less likelihood of cutting-off the

global optimum. On the other hand the most recent AP/OA/ER variant by Viswanathan and Grossmann (1989) has shown a remarkable degree of robustness to nonconvexities comparable to GBD. For instance all 18 problems in Table 1 converged to the global optimum despite the fact that half of these problems involve nonconvexities.

Finally, some other issues or observations that have emerged in the numerical solution are the following. A desirable feature in the OA algorithms is that the solution of NLP subproblems can be made successively easier to solve by using as a starting point the continuous variables predicted by the master problem (see Kocis and Grossmann, 1989). This follows from the fact that as iterations proceed the master problem becomes an increasingly better approximation of the MINLP problem. This feature cannot be exploited in GBD since its master problem does not involve continuous variables. In the case of branch and bound one can also obtain good guesses for the NLP from the solution of the previous node, although if the relaxation gap is large, the quality of the guesses for the initial nodes will not always be very good. As for the solution of the MILP master problems, the requirements by the OA algorithms will be the highest. In fact, for larger problems we have found that unless one resorts to an advanced MILP package (e.g. MPSX, MPSIII) these problems are prone to failure due to the accuracy that is required for the function linearizations, and to the more effective pruning techniques and features (e.g. special ordered sets) that are needed for large number of 0-1 variables. Future developments in cutting plane techniques (Crowder *et al.* 1983, Van Roy and Wolsey, 1987) may offer the possibility of solving with reasonable expense large MILPs with poor relaxation. Alternatively, recent search techniques (e.g. tabu search, Glover (1988), simulated annealing (Aaarts and van Laarhoven, 1985), neural networks (Carpenter and Grossberg, 1988)) could be used instead of solving directly large MINLP problems with poor relaxation.

SOLUTION STRATEGIES FOR MINLP IN PROCESS SYSTEMS

One obvious approach to the MINLP optimization of process flowsheets is to formulate the problem and solve it directly with any of the algorithms discussed in the previous section. A number of successful applications have been reported in the literature using such an approach (Kocis and Grossmann, 1987, 1988; Floudas and Paules, 1988; Yee and Grossmann, 1988; Ciric and Floudas, 1988b; Duran and Flores, 1988; Wellons and Reklaitis, 1989). However, 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 systems. Up to this date, not much work has been done in this area. Below we briefly describe a recent modelling-decomposition (M/D) strategy that has been proposed by Kocis and Grossmann (1988b) and which is especially suitable for heterogeneous networks where there is a one-to-one correspondence between units and tasks. Its major motivation has been to simplify the solution of the NLP and MILP problems, and to reduce the undesirable effect of nonconvexities and of having to optimize "dry units" with zero flows which are temporarily turned off in the superstructure. The solution of the NLP is simplified by optimizing only the particular flowsheet

at hand, as opposed to optimizing it within the superstructure as implied by problem (9). The MILP solution is simplified by only incorporating at each iteration an approximation to the particular flowsheet. Finally, the effect of nonconvexities is reduced by special modelling techniques.

The basic idea in the M/D strategy is to first recognize that a flowsheet superstructure can be viewed as a network consisting of two types of nodes: interconnection nodes (splitters and mixers) and process unit nodes (reactors, separators). In summary, the modelling is then performed as follows. Since interconnection nodes play a crucial role in defining the flowsheet structures and they exhibit well defined equations, special modelling techniques can be applied to these nodes. In particular, splitters and mixers that imply the choice of one single alternative can in fact be modelled through linear constraints which avoid the nonconvexities associated to the use of split fractions. For the case when multiple choices are possible, one can in fact develop valid linear outer-approximations that properly bound the nonconvex solution space in the MILP master problem. As for the process unit nodes, the mass balances are expressed in terms of component flows rather than in terms of fractional compositions. Lastly, the right hand side in the linearizations of the process units are modified to ensure that nonzero-flows are attained when the 0-1 variable is set to zero.

As for the decomposition part of this strategy the idea is as follows. Suppose we start by optimizing a particular flowsheet structure. It is clear that for the existing process units we are able to obtain linear approximations for the master problem. The question is then how to generate linear approximations of the "deleted" units in the superstructure. This can actually be accomplished by suboptimizing groups of units that are tied with existing interconnection nodes. Since prices (i.e. multipliers) and nonzero flows are available at these nodes, these can be used to suboptimize the nonexisting units "as if they were to exist" in the superstructure. This then provides not only nonzero flow conditions, but also points that are often very good for approximating these units. An example of how a superstructure based on an initial flowsheet can be decomposed into subsystems to be suboptimized is illustrated in Fig. 9 for the superstructure in Fig. 6. In this way, by optimizing the initial flowsheet structure, and suboptimizing the groups of nonexisting units, it then simply suffices to optimize the specific flowsheet that is generated at subsequent iterations in order to update the MILP. This then has two desirable effects: to only solve the NLP for each specific flowsheet, and to reduce the size of the MILP since only linearizations of existing units are incorporated at each iteration. This strategy is currently being automated in the flowsheet synthesizer PROSYN by Kravanja and Grossmann (1989), where the heat integration is handled through an extension of the constraints proposed by Duran and Grossmann (1986c) where area optimization is accounted for.

Another important aspect, is that up to date most of the results that have been reported in the literature are for systems where the MINLP is represented in equation form using simplified models. To our knowledge, the work by Harsh *et al* (1988) is the first where an MINLP procedure has been interfaced with a process simulator with rigorous models. These authors

implemented the OA algorithm in FLOWTRAN for the optimal retrofit design of flowsheets with fixed topology. For the case when the topology is also to be optimized, the M/D strategy could be used to circumvent the problem of simulating units with zero flows in a superstructure. It should also be noted that recently Duran and Flores (1988) have developed MINLP models for synthesizing heat integrated distillation sequences using the OA algorithm with rigorous thermodynamic models.

Finally, an important comment on solution strategies is the global approach for attacking the problem through MINLP or through related mathematical programming techniques. The strategy described above is still based on the idea of a simultaneous solution procedure where decomposition is being exploited by the problem structure. It is clearly conceptually sound to use a simultaneous approach since this reduces the risk of distorting trade-offs as interactions are explicitly accounted for (see also Grossmann, 1985).

On the other hand, it is always tempting to perform a sequential decomposition where instead the problem is sequentially partitioned according to a hierarchy of goals. A very good example of this is the synthesis procedure for heat exchanger networks that was proposed by Floudas *et al* (1986). Here, the problem was decomposed by minimizing the utility cost first with an LP, predicting the units with their stream matches by minimizing the number of units with an MILP, and finally optimizing with NLP a superstructure where the units are known but not their interconnections. This strategy, which was implemented in MAGNETS, has proved to be in general quite efficient. However, it has also had limitations, one of them being that due to the sequential decomposition suboptimal solutions can be obtained, especially when extended to multiperiod problems (Floudas and Grossmann, 1987; Gautam *et al*> 1988). The lesson to be learned here is that "insight" driven decomposition has significant risks of producing suboptimal solutions. However, at the same time that does not mean that this approach should be abandoned. For instance, in the heat exchanger synthesis problem Gundersen and Grossmann (1988) and Colberg and Morari (1989) have shown that by modifying the MILP step so as to anticipate the effects of area (e.g. using considerations of vertical heat transfer), the problem of obtaining suboptimal solutions is greatly reduced. Thus, the conclusions would seem that when performing decomposition the clue is to develop aggregate models that can to a large extent anticipate the decisions that are made downstream in a design. Good examples of this are the simultaneous optimization and heat integration method by Duran and Grossmann (1986c), and the compact scheduling models developed by Birewar and Grossmann (1989) that can effectively anticipate the effect of sequencing of batches at the design stage in multiproduct batch plants.

APPLICATIONS

Over the last 5 years MINLP optimization models have been reported for the synthesis of process flowsheets, heat exchanger networks, separation sequences, utility plants, and design of batch processes. Rather than describing in detail each of these works, we will briefly highlight

three examples from our research group at Carnegie Mellon to illustrate the capabilities and the current limitations of the MINLP approach.

Firstly, Kocis and Grossmann (1988b) have recently synthesized the HDA process that has been studied extensively by Douglas (1988). The superstructure for this process (heterogeneous network with one-to-one relationship between units and tasks) is shown in Fig. 10a which was derived based on a preliminary qualitative analysis of alternatives described in Douglas (1988). Given the basic options considered for selection of reactors, use of membrane separators and absorbers, and a restricted set of alternatives for the separation and recycle it is a relatively simple task to develop the superstructure representation. In this case the simplified nonlinear models were used to model the problem as an MINLP, which involves 13 0-1 variables, 672 continuous variables, and 678 constraints (140 nonlinear equations, 567 linear equations, 71 linear inequalities). The optimal solution, which is shown in Fig. 10b, was obtained with both the M/D strategy and with the AP/OA/ER algorithm on the full MINLP using MINOS and MPSX. The M/D strategy required 2 min of CPU time (DSM-3083), while AP/OA/ER required 8 min; both took 2 major iterations. This then shows the desirability of developing strategies that can exploit the structure of flowsheets. Furthermore, the example also shows how a qualitative pre-screening can lead to MINLP problems that are of reasonable size.

Secondly, Yee and Grossmann (1988) have developed a new superstructure representation for heat exchanger networks (see Fig. 4) which is particularly suitable for retrofit design (homogeneous network with one-to-many relations of units to tasks). 0-1 variables are used to denote the assignment of matches to units, and existence of pipe connections for the streams. A small example is shown in Fig. 11, where the optimal retrofit of the existing network (Fig. lla) is shown in Fig. lie. The solution that is obtained is rather unexpected and different from the intutitive solution in Fig. lib. Note that in Fig. 1 lc the bottom stream of the column is assigned to the cooler, and the reboiler of the column is reassigned for cooling purposes! This MINLP which involves 30 0-1 variables, 74 continuous variables and 144 constraints, was solved with the AP/OA/ER algorithm and with GBD using transshipment constraints to tighten the master problem. The AP/OA/ER algorithm required 100 sec (IBM-3083) and 3 major iterations, while GBD required 240 sec and 17 major iterations. The explanation for the slow convergence is the fact **that the** relaxed NLP solution had an objective value of only $3,777 compared to the $29,300 for **the** MINLP solution. Also while GBD required more iterations, its master problem took on average 8 sec per iteration versus 27 sec for the AP/OA/ER algorithm. Thus, this example shows that richer superstructures may exhibit poor NLP relaxations and consequently require considerable computational effort.

Lastly, Viswanathan and Grossmann (1989) have developed an MINLP model for determining the optimal feed tray location and the number of plates in distillation columns. The superstructure which is shown in Fig 12, consists of a number of potential trays, with subsets of them having potential feeds and by-pass streams to the condenser and reboiler (homogeneous network with one-to-one relation of units to tasks). The specific example that was considered

involved the separation of benzene and toluene and was modelled tray by tray with ideal thermodynamic correlations. For the case when a fixed number of 26 trays was specified, the MINLP involved 10 0-1 variables, 238 continuous variables and 239 constraints. The optimal feed tray location from among 10 candidate plates was determined by solving the relaxed NLP in 19 sec (IBM-3083). For the case when the number of trays was optimized for a fixed feed tray location from among 30 candidate trays, the MINLP involved 30 0-1 variables, 338 continuous variables and 467 constraints. In this case, the AP/OA/ER algorithm required 4 major iterations and 103 sec (IBM-3090). This example then shows that MINLP methods can be applied to complex process models.

CONCLUDING REMARKS

In this paper we have tried to present an overview of MINLP strategies and algorithms for process synthesis. From this review, it is clear that there has been considerable progress with this approach over the last few years. At the FOCAPD meeting in 1980, where the first session on process synthesis took place, heuristics and thermodynamic targets were dominant due to the skepticism and little hope that there was with the mathematical programming approach. At this point in time, however, we have evolved to a state where the modelling and solution of large-scale MINLP problems for synthesis can no longer be regarded as a Utopia. Results by our group at Carnegie Mellon, and by other researchers clearly support this claim.

While one can expect that the scope of MINLP optimization for synthesis will increase, it is also clear that a number of important challenges remain unsolved, and which most likely will be the subject of future work. Below we cite few major open questions:

1. How to systematically develop superstructures for heterogeneous systems?
2. How to effectively solve MBLP and MINLP problems with poor relaxations?
3. What is the likely impact on MINLP of recent algorithmic developments such as Karmarkar's algorithm or strong cutting planes techniques?
4. What is the role of the new generation of combinatorial search techniques for synthesis such as neural networks, simulated annealing and tabu search?
5. How to effectively exploit the computational power of parallel computers to increase by several orders of magnitude the size of MINLP problems that can currently be solved?
6. How to effectively combine and integrate the MINLP optimization paradigm with qualitative AI techniques?

Although these questions remain largely unanswered at this point, there is no doubt that over the next decade we will see some exciting developments along these lines.

ACKNOWLEDGMENT

The author would like to acknowledge financial support from the National Science Foundation under Grant CPE-8351237, and for partial support from the Engineering Design Research Center at Carnegie Mellon University.

REFERENCES

Aarts, E.H.L. and van Laarhoven, P.J.M. (1985). Statistical Cooling: A General Approach to Combinatorial Optimization Problems. *Philips J. Res.* 40,193.

Achenie, L.K. and Biegler, L.T. (1986). Algorithmic Synthesis of Chemical Reactor Networks Using Mathematical Programming. *Industrial and Engineering Chemistry Research* 25,621.

Andrecovich, MJ. and Westerberg, A.W. (1985). An MILP Formulation for Heat-Integrated Distillation Sequences Synthesis. *AIChE Journal* 31(9), 1461-1474.

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

Beltramini, L.J., Sheppard, CM. and Motard, R.L. (1989). Truth Maintenance Systems in Process Synthesis. Paper 31c, National AIChE Meeting, Houston.

Benders, J.F. (1962). Partitioning Procedures for Solving Mixed-variables Programming Problems. *Numerische Mathematik* 4,238-252.

Birewar, D.B. and Grossmann, I.E. (1989). Incorporating Scheduling in the Optimal Design of Multiproduct Batch Plants. *Computers and Chem. Eng.* 13,141.

Brooke, A., Kendrick, D. and Meeraus, A. (1988). GAMS A User's Guide, Scientific Press, Palo Alto.

Carpenter, G. and Grossberg, S. (1988). The ART of Adaptive Pattern Recognition by a Self-Organizing Neural Network'[1], *IEEE Computer* 21,77-88.

Cavalier, T.M. and Soster, A.L. (1987). Logical Deduction Via Linear Programming. IMSE Working Paper 87-147, Department of Industrial and Management Systems Engineering, Pennsylvania State University.

Cerda, J. and Westerberg, A.W. (1983). Synthesizing Heat Exchanger Networks Having Restricted Stream/Stream Matches Using Transportation Problem Formulations, *Chemical Engineering Science* 38,1723-1740.

Chitra, SP. and Govind, R. (1985). Synthesis of Optimal Serial Reactor Structures for Homogeneous Reactions. *AIChE Journal* 31,177-193.

Ciric, A.R. and Floudas, C.A.(1988a). Global Optimum Issues on Heat Exchanger Network Synthesis. *Proceeding ofPSE'88*,104, Sidney.

Ciric, A.R. and Floudas, C.A. (1988b). Simultaneous Optimization of Process Stream Matches and Heat Exchanger Formulations. Paper No. 79e, Annual AIChE Meeting, Washington, DC.

Colberg, R.D. and Morari, M. (1989). Area and Capital Cost Targets for Heat Exchanger Network Synthesis with Constrained Matches and Unequal Heat Transfer Coefficients. Submitted to *Computers and Chem. Eng.*.

Colmenares, T.R. and Seider, W.D. (1987). Heat and Power Integration of Chemical Processes. *AIChE Journal* 33, 898.

Crowder, H., Johnson, E.L. and Padberg, M. (1983). Solving Large-Scale Zero-One Linear Programming Problems. *Operations Research* 31(5), 803-834.

Dolan, W.B., Cummings, P.T. and Levan, M.D. (1987). Heat Exchanger Network Design by Simulated Annealing. *Proceedings ofFOCAPO Meeting* (G.V. Reklaitis and H.D. Spriggs, eds.), 639-645, Elsevier.

Douglas, J.M. (1985). A Hierarchical Decision Procedure of Process Synthesis. *AIChE Journal* 31(3), 353-362.

Douglas, J. M. (1988). Conceptual Design of Chemical Processes", McGraw-Hill, New York.

Duran, M.A. and Grossmann, I.E. (1986a). A Mixed-Integer Nonlinear Programming Approach for Process Systems Synthesis. *AIChE Journal* 32(4), 592-606.

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

Duran, M.A. and Grossmann, LE. (1986c). Simultaneous Optimization and Heat Integration of Chemical Processes" *AIChE Journal* 1,123-138.

Duran, M.A. and Flores, A. (1988). Mixed-Integer Nonlinear and Linear Programming Approaches to the Synthesis of Heat Integrated Distillation Sequences. Paper No. 81c, Annual AIChE Meeting, Washington, D.C.

Eliceche, A.M. and R.W.H. Sargent (1986). Synthesis and Design of Distillation Sequences. *LChem£. Symposium Series No. 61*,1-22.

Floudas, C.A., Ciric, A.R. and Grossmann, LE. (1986). Automatic Synthesis of Heat Exchanger Networks. *AIChE Journal* 32(2), 276-290.

Floudas, C.A. and Grossmann, I.E. (1987). Synthesis of Flexible Heat Exchanger Networks with Uncertain Flowrates and Temperatures. *Computers and Chem. Eng.* 11,319.

Floudas, C.A. and Anastasiadis, S.H. (1988). Synthesis of General Distillation Sequences with Several Multicomponent Feeds and Products. *Chemical Engineering Science* 43,2407.

Floudas, C.A. and Paules, G.E. (1988). A Mixed-Integer Nonlinear Programming Formulation for the Synthesis of Heat Integrated Distillation Sequences. *Computers and Chem. Eng.* 12(6), 531-546.

Floudas, C.A., Aggarwal, A. and Ciric, A.R. (1988). Global Optimum Search for Nonconvex NLP and MINLP Problems. Paper No. 76c, Annual AIChE Meeting, Washington, DC.

Floquet, P., Pibouleau, L. and Domenech, S. (1988). Mathematical Programming Tools for Chemical Engineering Process Design Synthesis. *Chemical Engineering and Processing* 23(2), 99.

Garfinkel, R. S., Nemhauser, G. L. (1972). Integer Programming. John Wiley and Sons, New York.

Gautam, R., H.S. Chen and J.S. Warec (1988). Designing Flexible Heat Exchanger Networks. Paper 39 f, National AIChE Meeting, New Orleans.

Geoffrion, A.M. (1972). Generalized Benders Decomposition. *Journal of Optimization Theory and Applications"* 10(4), 237-260.

Glasser, D., Crowe, C. and Hildebrandt, D. (1987). A Geometric Approach to Steady Row Reactors: the Attainable Region and Optimization in Concentration Space. *Industrial and Engineering Chemistry Research* 26,1803.

Glover, F. (1986). Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers and Optns. Res.* 13,533-549.

Glover, F. (1988). Tabu Search. Center for Applied Artificial Intelligence, CAAI Report 88-3, Boulder, Colorado.

Gomez, M.A. and J.D. Seader (1976). Separator Sequence Synthesis by a Predictor Based Ordered Search.A/C7iE *Journal!!,* 970.

Grossmann, I.E. (1985). Mixed-Integer Programming Approach for the Synthesis of Integrated Process Flowsheets. *Computers and Chemical Engineering* 9(5), 463-482.

Gundersen, T. and Grossmann, IJE. (1988). Improved Optimization Strategies for Automated Heat Exchanger Network Synthesis Through Physical Insights. Paper No. 81 g, AIChE Meeting, Washington, D.C.

Gundersen, T. and Naess, L. (1988). The Synthesis of Cost Optimal Heat Exchanger Networks-An Industrial Review of the State of the Art. *Computers and Chem. Eng.* 12,503-530.

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 Optimization Theory and Applications* 22,297.

Harsh, M.G., Saderne, P. and Biegler, L.T. (1988). A Mixed Integer Flowsheet Optimization for Process Retrofits: The Debottlenecking Problem. Submitted to *Computers and Chemical Engineering* (1988).

Hendry, J£. and Hughes, R.R. (1972). Generating Separation Process Flowsheets. *Chem. Eng. Progress* 68,69.

Hendry, J.E., Rudd, DJ[7]. and Seader, J.D. (1973). Synthesis in the Design of Chemical Processes. *AIChE Journal* 19,1.

Hoffman, K. and Padberg, M. (1986). LP-Based Combinatorial Problem Solving. *Annals of Operations Research,* 4,145-193.

Hohmann, E.C. (1971). Optimum Networks for Heat Exchange. Ph.D. Thesis, University of Southern California.

IBM Mathematical Programming System Extended/370 (MPSX/370), Basic Reference Manual, White Plains, NY (1979).

Jackson, R. (1962). Optimization of Chemical Reactors with Respect to Row Configuration. *Journal of Optimization Theory and Applications* 2,240-259.

Jones, S.A. and D.W.T. Rippin (1985). The Generation of Heat Load Distributions in Heat Exchanger Network Synthesis. *Proceedings of PSE'85*, 157-177, Cambridge (U.K.).

Karmarkar, N. (1984). A New Polynomial-Time Algorithm for Linear Programming. *Combinatorica*, 4(4), 373-395.

Kocis, G.R. and Grossmann, I.E. (1987). Relaxation Strategy for the Structural Optimization of Process Flowsheets. *Industrial and Engineering Chemistry Research* 26(9), 1869-1880.

Kocis, G.R. and Grossmann, I.E. (1988a). Global Optimization of Nonconvex MINLP Problems in Process Synthesis. *Industrial and Engineering Chemistry Research* 27, 1407-1421.

Kocis, G.R. and Grossmann, I.E. (1988b). A Modelling/Decomposition Strategy for MINLP Optimization of Process Flowsheets. Paper No. 76a, AIChE Meeting, Washington, D.C.

Kocis, G.R. and Grossmann, I.E. (1989). Computational Experience with DICOPT Solving MINLP Problems in Process Synthesis Engineering. *Computers and Chem. Eng.* 13, 307-315.

Kravanja, Z. and I.E. Grossmann, "PROSYN - An MINLP Process Synthesizer", manuscript in preparation (1989).

Linnhoff, B. and Flower, J.R. (1978). Synthesis for Heat Exchanger Networks. I. Systematic Generation of Energy Optimal Networks. *AIChE Journal* 24, 633.

Linnhoff, B. and Hindmarsh, E. (1983). The Pinch Design Method of Heat Exchanger Networks. *Chemical Engineering Science* 38 745.

Magnanti, M.L. and Wong, R.T. (1981). Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria. *Operations Research*, 29, 464-484.

Mahalec, V. and Motard, R.L. (1977). Evolutionary Search for an Optimal Limiting Process Flowsheet. *Computers and Chem. Eng.* 1, 149.

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

Mawengkang, H. and Murtagh, B.A. (1986). Solving Nonlinear Integer Programs with Large-Scale Optimization Software. *Annals of Operations Research*, 5, 425-437.

Mawengkang, H. (1988). Nonlinear Integer Programming. Ph.D. thesis, The University of New South Wales, Australia.

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

Nemhauser, G.L. and Wolsey, L.A. (1988). Integer and Combinatorial Optimization. Wiley-Interscience, New York.

Nishio, M., Itoh, J., Shiroko, K. and Umeda, T. (1980). Thermodynamic Approach to Steam and Power System Design. *Ind. Eng. Chem. Process Des. Dev.* 19, 306.

NishidaJsL, Stephanopoulos,G.,and Westerberg, A. (1981). A Review of Process Synthesis. *AIChE Journal* 27, 321-351.

Ostrovsky, G.M., Ostrovsky, M.G. and Mikhailow, G.W. (1988). Discrete Optimization of Chemical Processes. Submitted to *Computers and Chem. Eng.*

Papoulias, S. and Grossmann, I.E. (1983). A Structural Optimization Approach in Process Synthesis, Parts I, II, and III. *Computers and Chem. Eng.* 7(6), 695-734.

Paules, G.E. and Floudas, C.A. (1989). APROS: A Discrete-Continuous Optimizer for Solution of Mixed-Integer Nonlinear Programming Problems. To appear in *Operations Research.*

Pekny, J. and Miller, D. (1989). A Parallel Branch and Bound for Solving Large Asymmetric Traveling Salesman Problems. Presented at CORS/ORSA/TIMS Meeting, Vancouver.

Powell, MJ.D. (1977). A Fast Algorithm for Nonlinearly Constrained Optimization Calculations. Presented at Dundee Conference on Numerical Analysis.

Ryan, PJ. and M.F. Doherty (1987). Synthesis and Optimal Design of Alternative Sequences for Separating Heterogeneous Azeotropic Mixtures. Paper 91b, Annual AIChE Meeting, New York.

Saboo, A.K., Morari, M. and Colberg, R.D. (1986). RESHEX:An Interactive Software Package for the Synthesis and Analysis of Resilient Heat-Exchanger Networks. *Computers and Chemical Engineering* 10(6), 577-600.

Sahinidis, N.V., Grossmann, I.E., Fornari, R.E., and Chathrathi, M. (1989). Optimization Model for Long Range Planning in the Chemical Industry. To appear in *Computers and Chem. Eng.*

Sahinidis, N.V. and Grossmann, I.E. (1989). MINLP Model for Scheduling in Continuous Parallel Production Lines. To be presented at AIChE Meeting, San Francisco.

Sargent, R.W.H. and Gaminibandara, K. (1976). Optimal Design of Plate Distillation Columns. *Optimization in Action* (L.W.C. Dixon, ed.), 267-314, Academic Press London.

Schrijver,A. (1986) Theory of Linear and Integer Programming. John Wiley, New York.

Shelton, M.R. and Grossmann (1986).Optimal Synthesis of Integrated Refrigeration Systems. I: Mixed-Integer Programming Model. *Computers and Chem. Eng.* 10(5), 445-459.

Siirola, JJ., Powers, GJ. and Rudd, D.F. (1971). Synthesis of Systems Design, in. Toward a Process Concept Generator. *Industrial and Engineering Chemistry Fundamentals* 17 677.

Simon, H.A. (1987). Two Heads Are Better Than One: The Collaboration Between AI and OR. *Interfaces* 17,8-15.

Stephanopoulos, G. and Westerberg, A. (1976). Studies in Process Synthesis. II. Evolutionary Synthesis of Optimal Process Flowsheets. *Chemical Engineering Science* 31,195-204.

Stephanopoulos, G. (1981). Synthesis of Process Flowsheets: An Adventure in Heuristic Design or a Utopia of Mathematical Programming?. *Foundations of Computer-Aided Chemical Process Design* (R.S.H. Mah and W.D. Seider, eds.), Engineering Foundation, New York, Vol. 2, pp 439-499.

Swaney, R.E. (1988). Thermal Integration of Processes with Heat Engines and Heat Pumps. Submitted for publication.

Townsend, D.W. and Linnhoff, B. (1983). Heat and Power Networks in Process Design. I. Criteria for Placement of Heat Engines and Heat Pumps in Process Networks. *AIChE Journal* 29 742-748.

Umeda, T., Hirai, A. and Ichikawa, A. (1972). Synthesis of Optimal Processing Systems by an Integrated Approach. *Chemical Engineering Science* 27,795-804.

Umeda, T., Harada, T. and Shiroko, K. (1979). A Thermodynamic Approach to the Synthesis of Heat Integration Systems in Chemical Processes. *Computers and Chem. Eng.* 3,273-282.

Van Roy, TJ. and Wolsey, L.A. (1987). Solving Mixed Integer Programs by Automatic Reformulation. *Operations Research* 35,45-57.

Vasantharajan, S., Viswanathan, J. and Biegler, L.T. (1989). Large Scale Development of Reduced Successive Quadratic Programming. Presented at CORS/TIMS/ORSA Meeting, Vancouver.

Viswanathan, J. and Grossmann, I.E. (1989). A Combined Penalty Function and Outer-Approximation Method for MINLP Optimization. Presented at CORS/TIMS/ORSA Meeting, Vancouver.

Wehe, R.R. and Westerberg, A. W. (1987). An Algorithmic Procedure for the Synthesis of Distillation Sequences with Bypass. *Computers and Chemical Engineering* 11(6), 619-627.

Wellons,H.S. and Reklaitis, G.V. (1989). The Design of Multi-Product Batch Plants Under Uncertainty with Staged Expansions. *"Computers and Chem. Eng.* 13(1/2), 115-126.

Westerberg, A.W. (1985). The Synthesis of Distillation Based Separation Sequences. *Computers and Chem. Eng.* 9, 421.

Williams, H.P. (1978). The Reformulation of Two Mixed Integer Programming Problems. *Mathematical Programming* 14, 325-331.

Yee, T.F. and Grossmann, I.E. (1988). A Screening and Optimization Approach for the Retrofit of Heat Exchanger Networks. Paper 8Id, Annual AIChE Meeting, Washington, DC.

Yee, TJ[7]. and Grossmann, IJE. (1989). Design and Synthesis of Multistream Heat Exchanger Networks. Manuscript under preparation.

Yuan, X., Zhang, S., Pibouleau, L. and Domenech, S. (1987). Une Methode d'optimisation non lineaire en variables mixtes pour la conception de procedes. Partie I: Presentation de l'agorithme. Submitted for publication as the RAIRO Recherche Operationnelle.

Table 1

Computational Results with DICOPT++

(Augmented-Penalty/Outer-Approximation/Equality-Relaxation)

Problem	Size MINLP 0-1	Cont.V.	Const.	Iterations*	Time** (sec)	% NLP:MILP
LAZIMI	2	8	5	1	0.22	100:00
HW74	3	9	9	4	2.43	26:74
NONCON	3	3	6	1	0.1	100:00
YUAN	4	4	10	3	1.87	36:64
CAPITAL	10	3	7	4	2.32	22:78
FLEX	4	12	16	3	2.24	46:54
REL1	16	21	18	3	17.9	36:64
EX3	8	26	32	5	3.7	51:49
EX4	25	7	31	5	65.1	9:91
BATCH	24	23	74	3	7.9	70:30
BATCH8	40	33	142	4	24.4	63:37
BATCH12	60	41	218	4	38.5	45:55
TABATCH	24	71	129	3	53.6	53:47
UTILRED	28	118	168	3	41.9	14:86
TFYHEN	30	74	144	3	102.0	20:80
EX5FEED	10	238	239	1	19.2	100:00
EX5TRAY	30	338	467	4	103.6	39:61
HDASS	13	709	719	3	482.0	84:16

* N iterations require N NLP subproblems and N-1 MILP master problems

" Total time NLP:MINOS / MILP:MPSX. All problems on IBM-3083, except EX5TRAY on IBM-3090

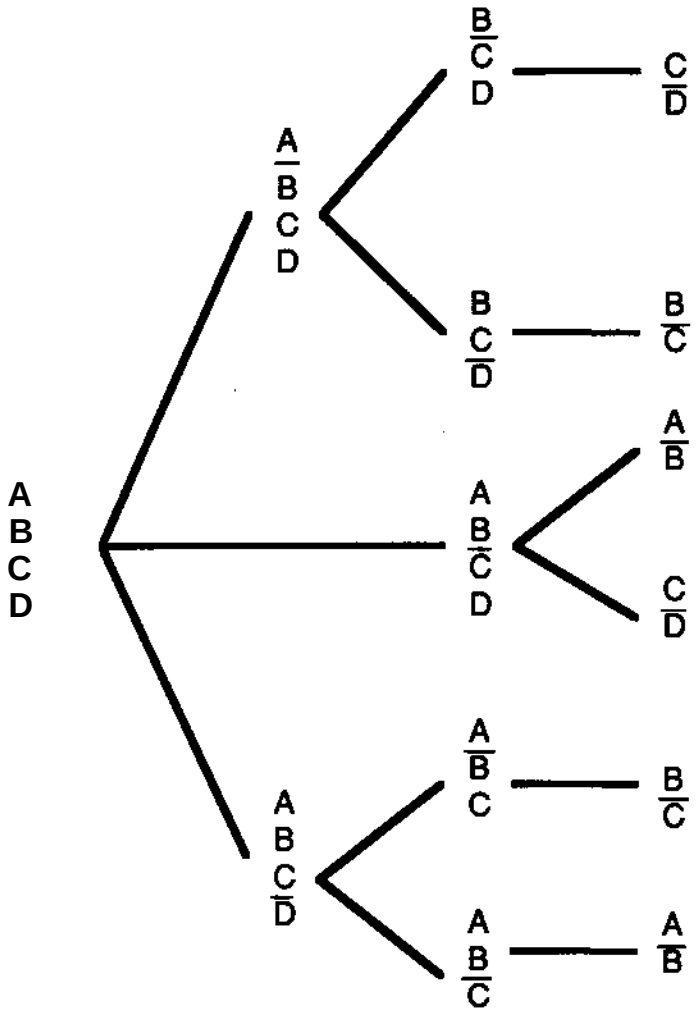


Fig. 1. Tree representation for separation of 4-component mixture.

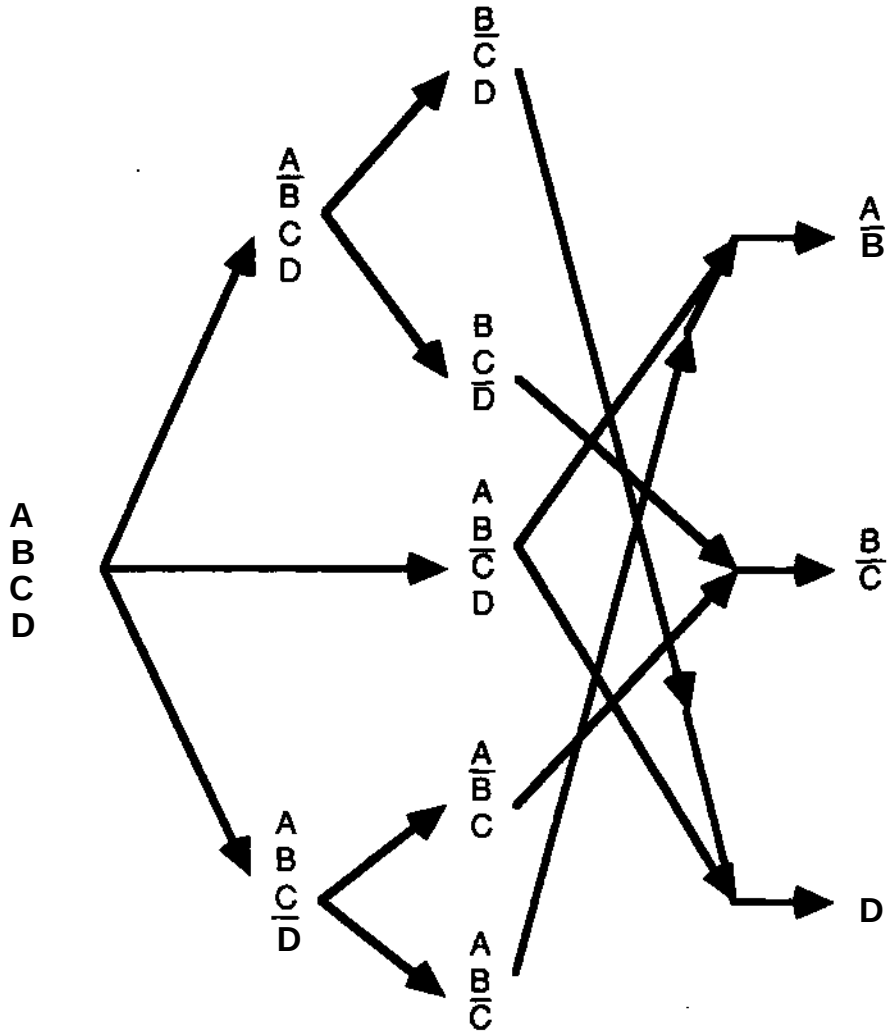


Fig. 2. Network representation for separation of 4-component mixture.

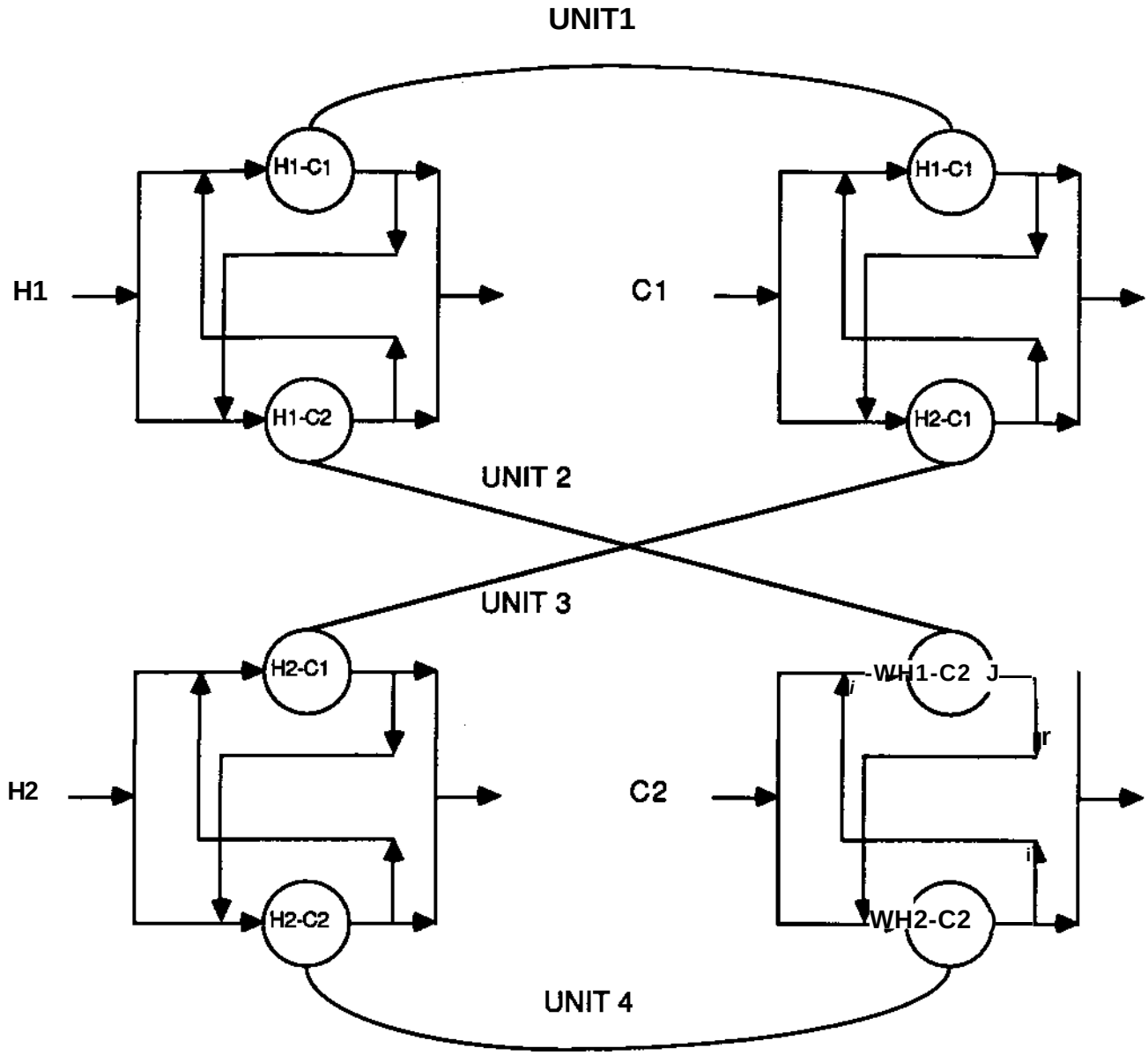


Fig. 3. Heat exchanger network superstructure by Floudas et al (1986).

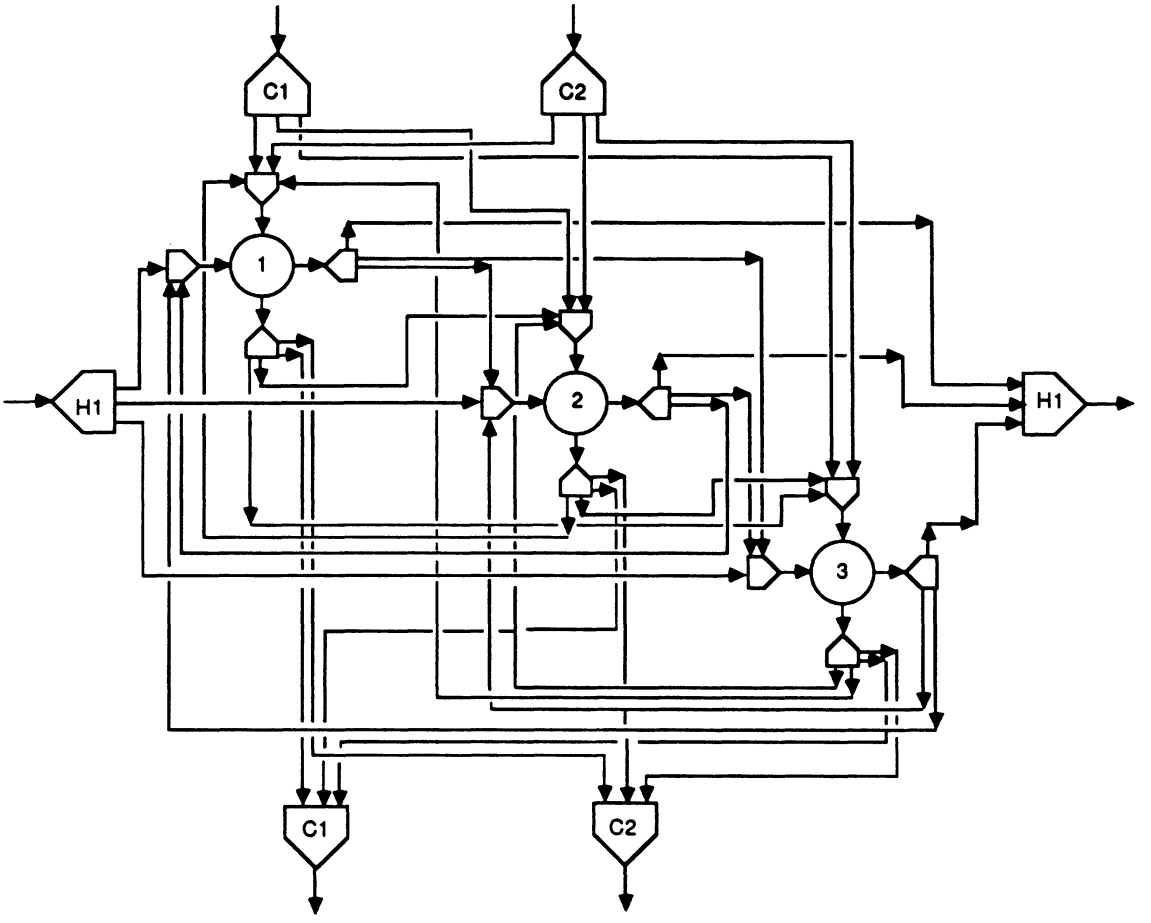


Fig. 4. Superstructure by Yee and Grossmann (1988) for heat exchanger network.

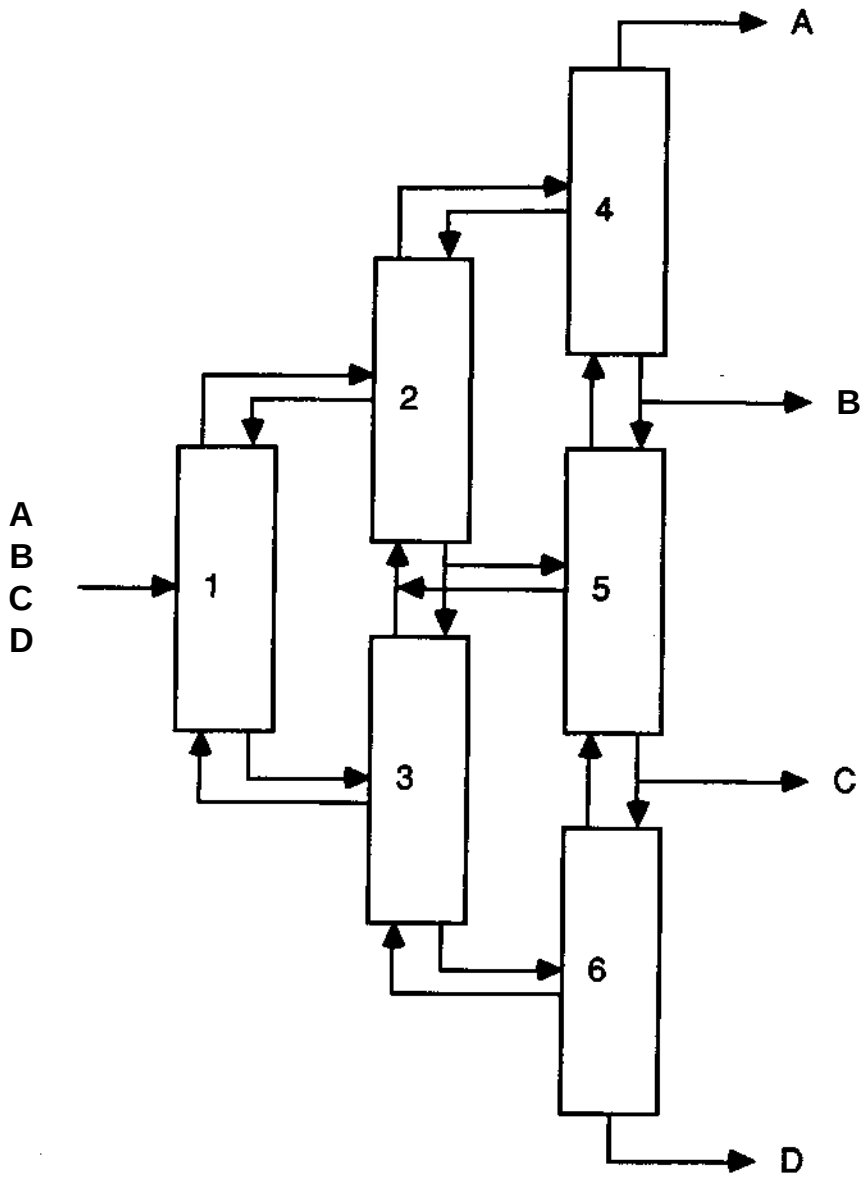


Fig. 5. Superstructure by Sargent and Gaminibandara (1976) for separation of 4 components.

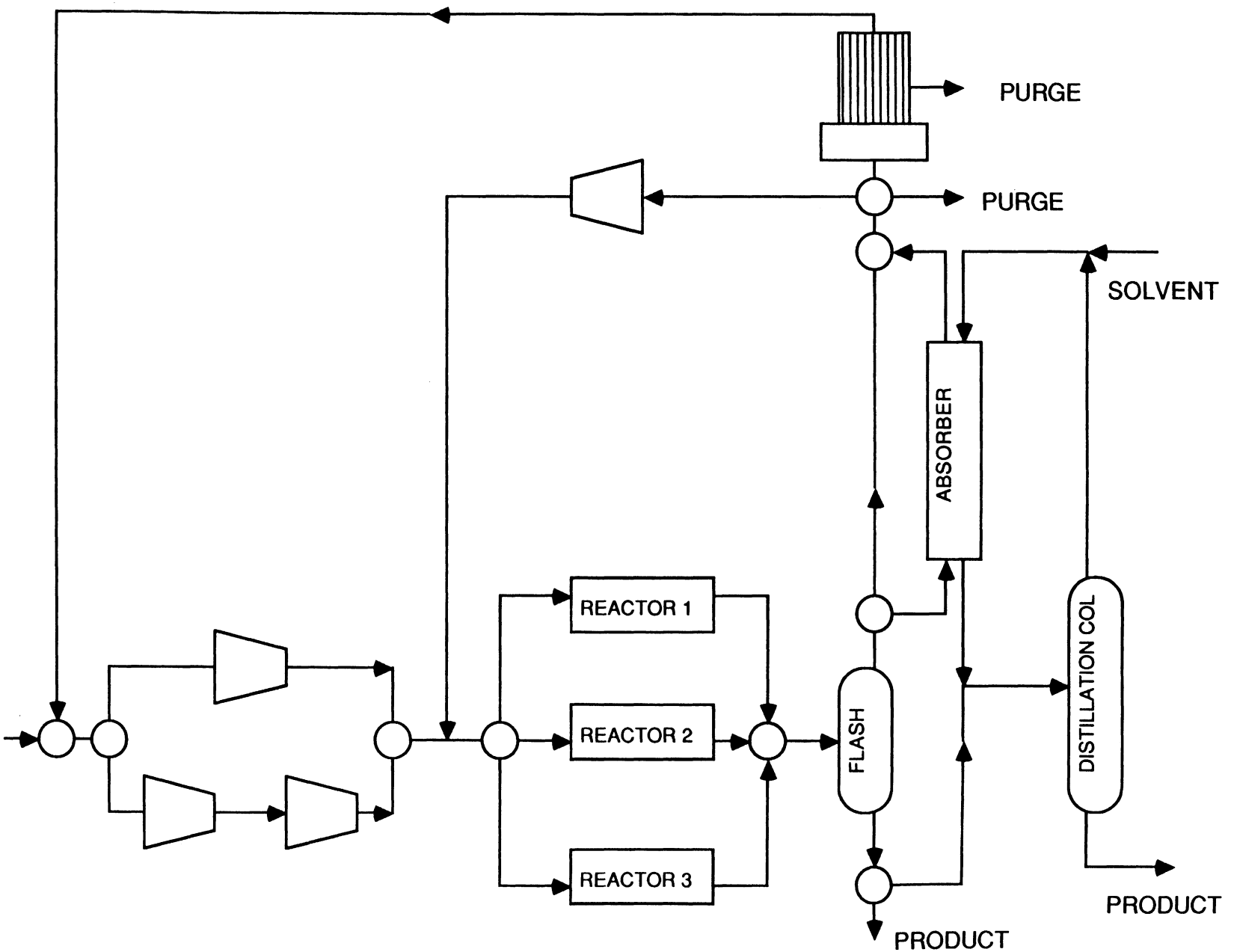
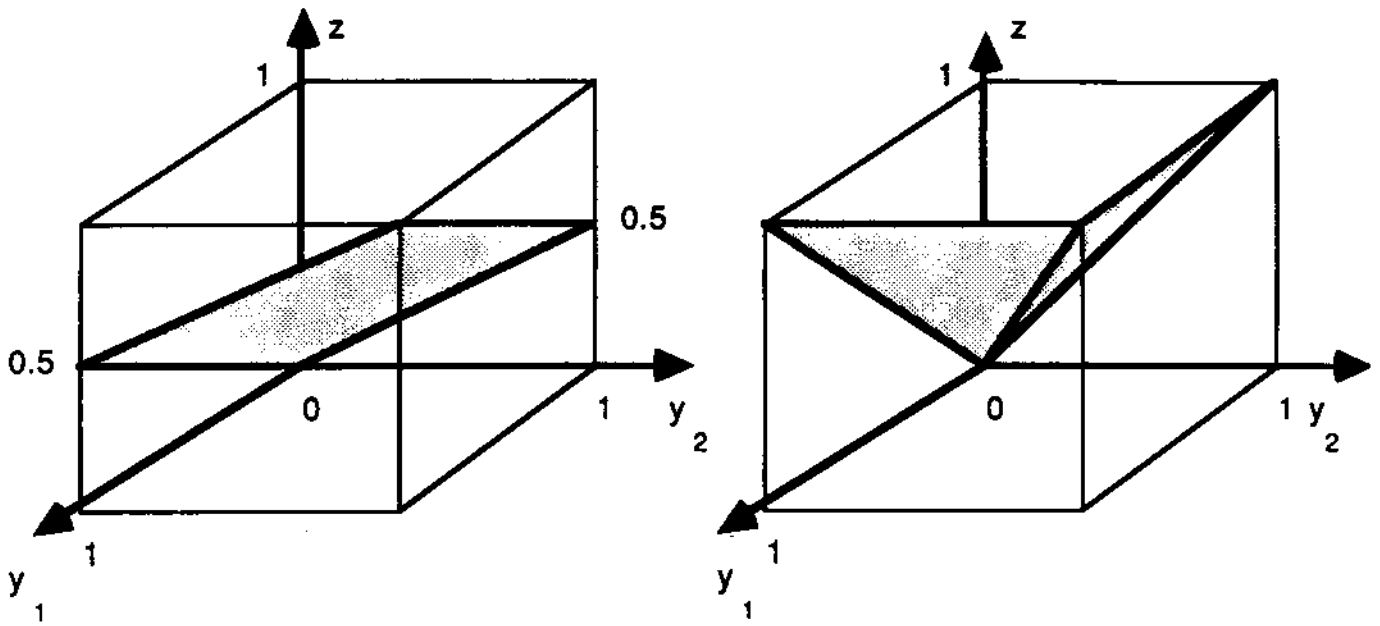


Fig. 6. Flowsheet superstructure for specific alternatives.



(a) $y_1 + y_2 - 2z \leq 0$

(b) $y_i - z \leq 0$
$y_2 - z \leq 0$

Fig. 7. Plot of feasible region for alternative constraint models.

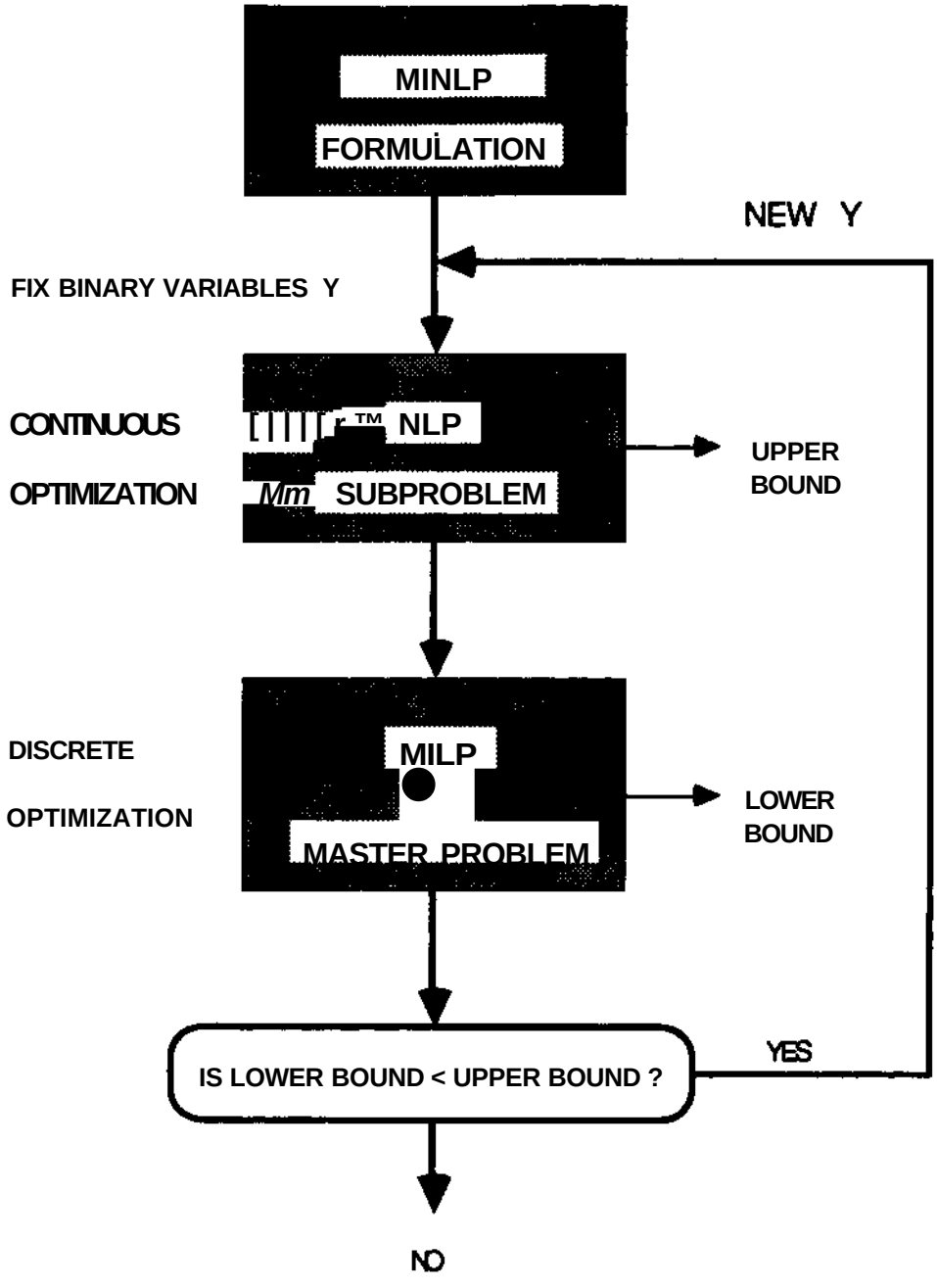
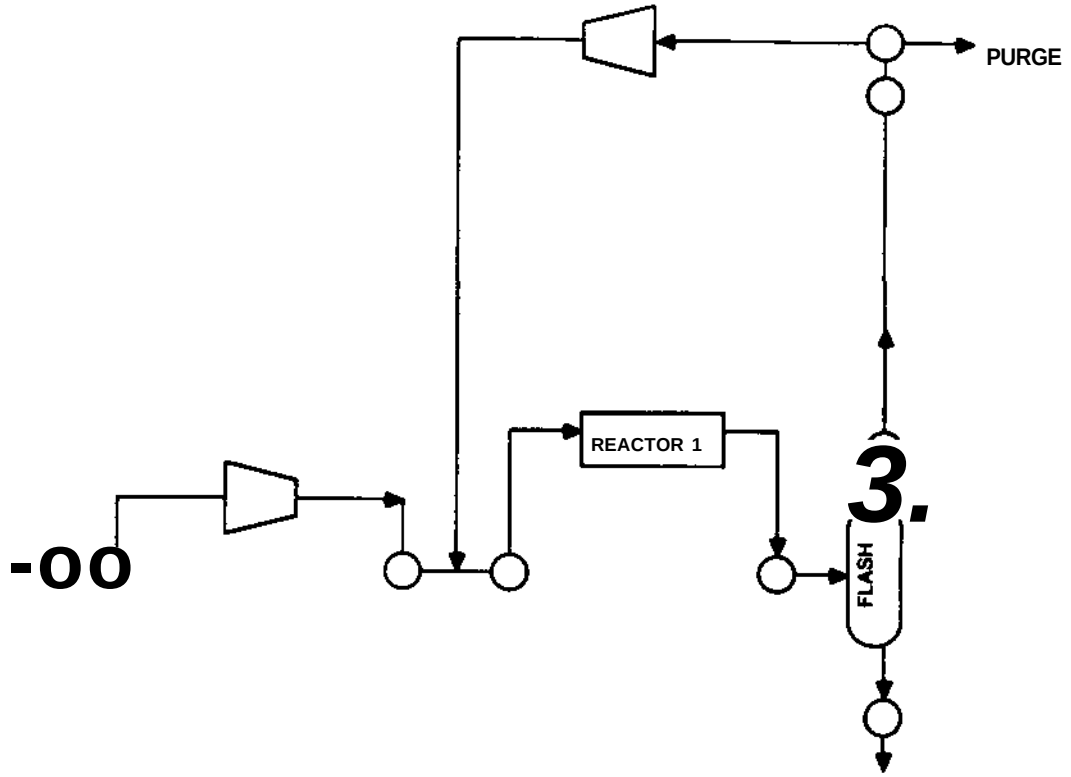
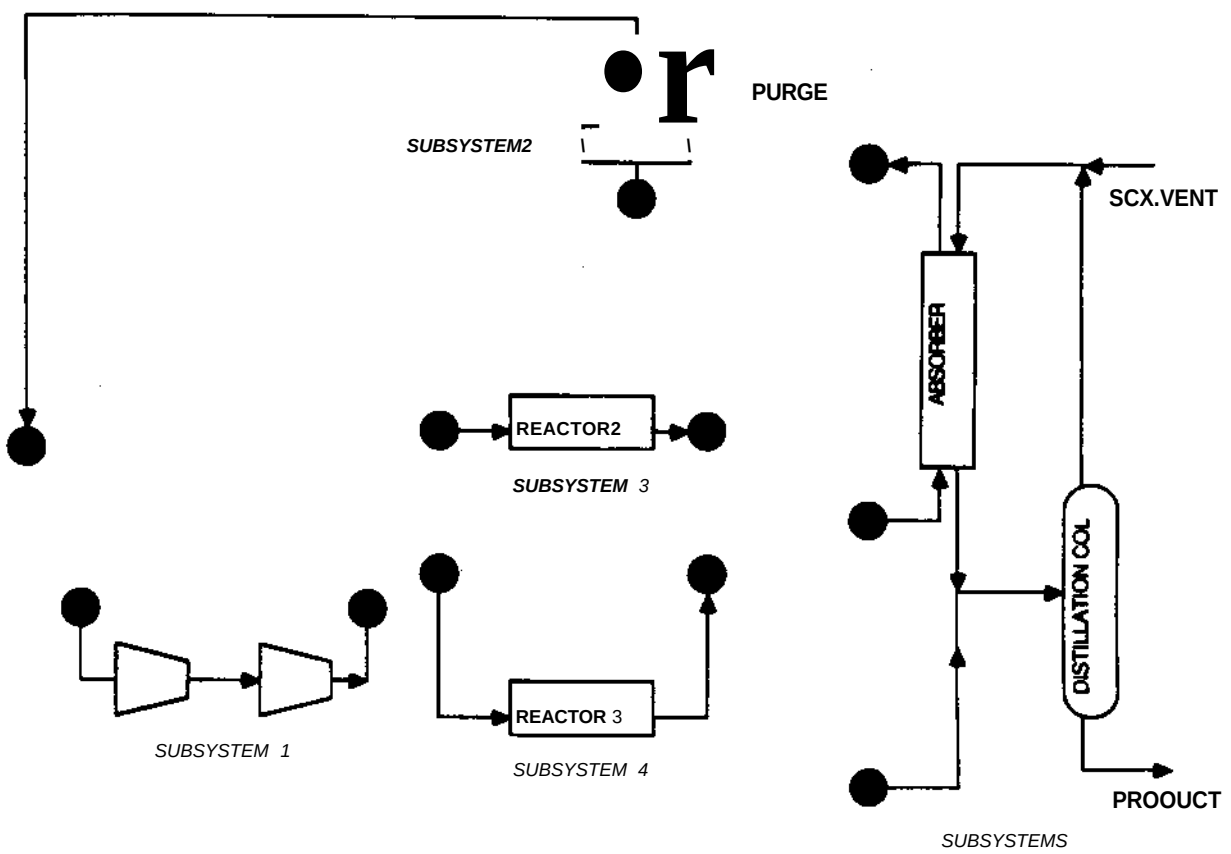


Fig. 8. Main steps in GBD and OA algorithms.

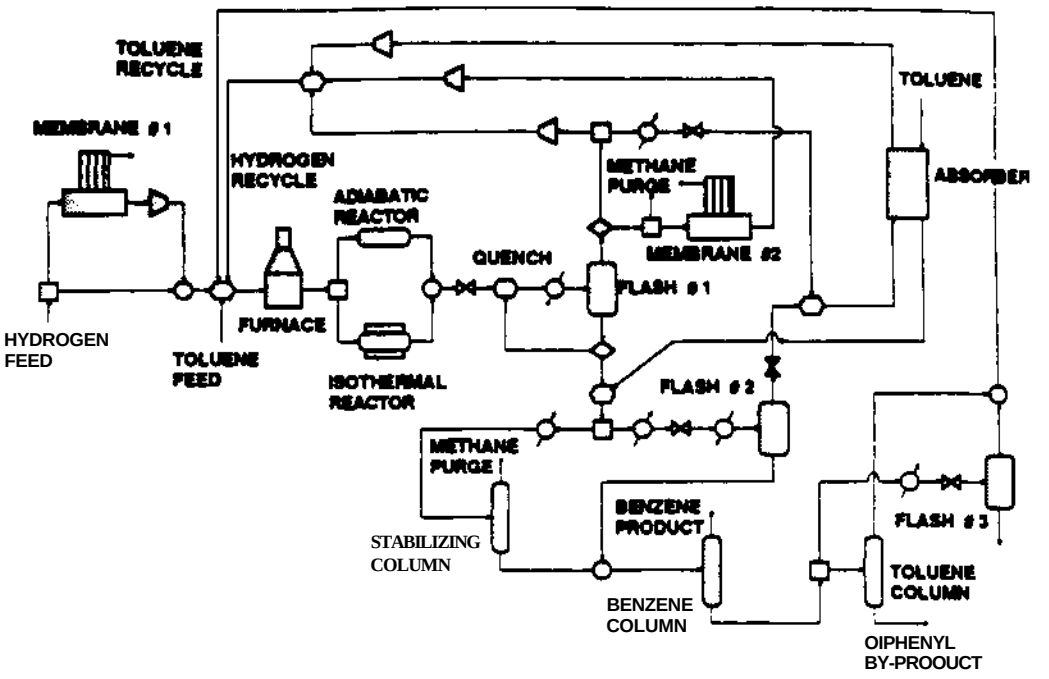


(a) Initial Flowsheet



(b) Subsystems for suboptimization

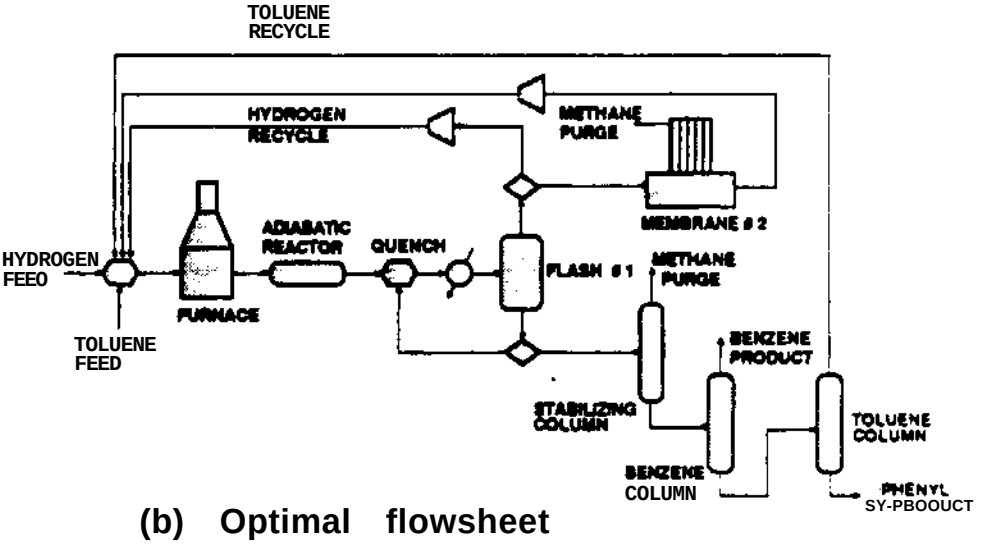
Fig. 9. Decomposition of flowsheet superstructure in Fig. 6.



LFCCND rflB INTtBCONNfCTION NOPff
O SNQLE CHOICE STREAM SPUTTER
O MULTIPLE CHOICE STREAM SPLITTER
O fINQLE CHOICE STREAM MUER
O MULTIPLE CHOICE STREAM MIXER

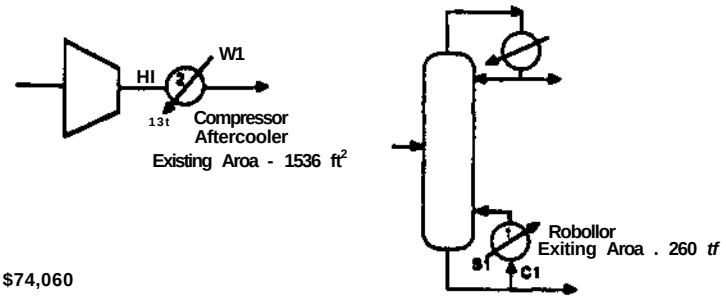
UOBLEM 1TATISTICS	
13	BMARY VARUBLES
•72	CONTVIUOUS VARUBLES
140	NONLINEAR CONSTRAINTS
S3B	UNEAR CONSTRAINTS

(a) Superstructure



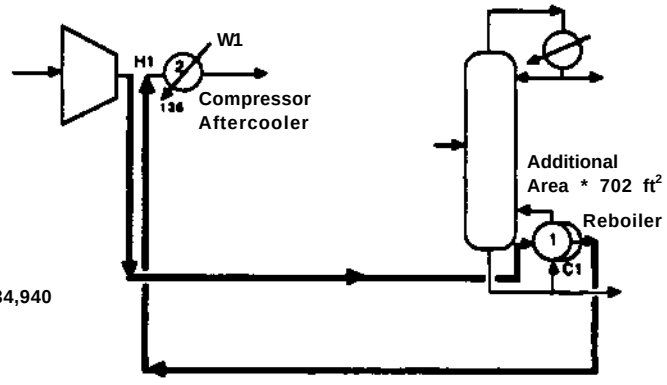
(b) Optimal flowsheet

Fig. 10. Superstructure optimization of HDA toluene process.



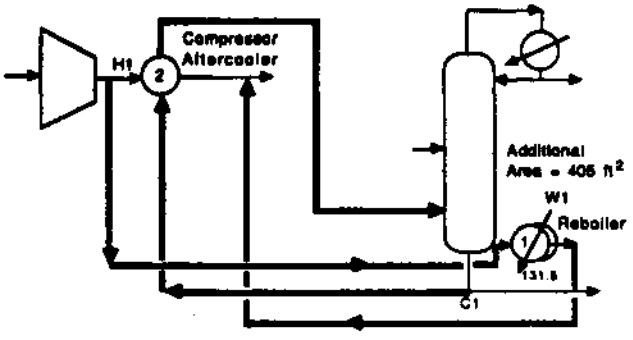
Annual Utility Cost » $74,060

EHistino Netmork



Annual Cost * $34,940

Intuitiue Solution



Annual Cost - $29,390

Retrofit Solution

Fig. 11. Retrofit optimization with superstructure by Yee and Grossmann (1988).

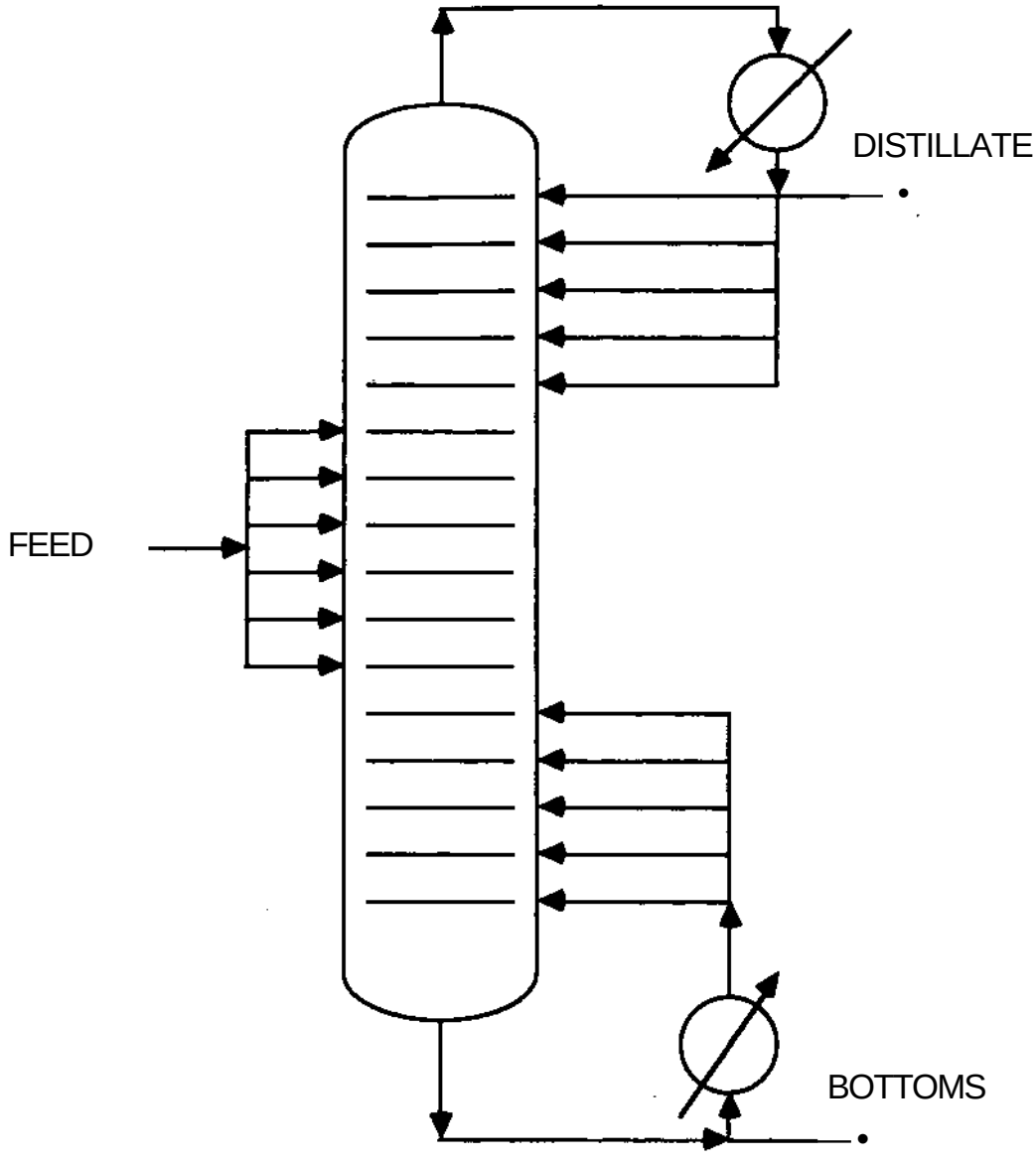


Fig. 12. Superstructure for feed tray location and number of plates.