MIXED-INTEGER OPTIMIZATION TECHNIQUES FOR ALGORITHMIC PROCESS SYNTHESIS

Ignacio E. Grossmann

Department of Chemical Engineering
Carnegie Mellon University
Pittsburgh, Pennsylvania

This paper presents an overview of the mathematical programming approach for process synthesis. First, the methods for process synthesis are reviewed with an emphasis on algorithmic methods. The mathematical programming approach is covered next in a discussion of basic concepts on representations for synthesis, modeling of mixed-integer nonlinear programming (MINLP) problems, MINLP algorithms, and solution strategies. As is shown, these four components are basic elements in the algorithmic methods for process synthesis. Also, it is shown, both through the derivation of methods and their application *to several examples, that MINLP optimization has reached a stage where it can solve practical problems of significant size. Finally, several future directions of research are also discussed.*

I. Introduction

A major step in process synthesis has been its formalization through the mathematical programming approach (Grossmann, 1990a,b; Grossmann and Daichendt, 1994). Major motivations include the increased need for the automation of the synthesis process and the need to explore a larger number of alternatives at the preliminary stages of design in order to improve the economics and other design criteria. In this paper we provide a comprehensive overview of the algorithmic approach to process synthesis, particularly the one that relies on the use of MINLP optimization.

This paper is organized as follows. We first present an extensive overview of previous work in process synthesis, emphasizing algorithmic methods. Next, we outline the major steps that are involved in the mathematical programming approach: representation of alternatives, mathematical modeling, and solution of the corresponding mathematical programming problem, which in general involves a mixed-integer nonlinear programming (MINLP) problem. For the former step, we discuss several major alternative representations that can be used, and clarify the use of representations at different levels of abstraction, particularly superstructures and aggregated models. We then concentrate on the modeling, highlighting the usefulness of propositional logic for deriving discrete constraints. Next, we give a unified presentation of the major MINLP algorithms that have emerged, emphasizing their common features. Then we discuss the role of solution strategies to address difficulties with zero flows, large-scale problems, and nonconvexities. Finally, several example problems are presented to illustrate some of the major points in this paper. The paper concludes with a discussion of future research directions.

II. Overview of Previous Work

A. Process Flowsheets

Reviews on earlier developments in the area of process synthesis can be found in Hendry *et al.* (1973), Hlavacek (1978), and Nishida *et al.* (1981). In the late sixties, work began to develop a systematic approach to process synthesis based mostly on the use of decomposition and heuristic rules (Rudd and

Watson, 1968; Rudd, 1968; Masso and Rudd, 1969). Algorithmic methods for selecting the optimal configuration from a given superstructure also began to be developed through the use of direct search methods for continuous variables (Umeda *et al.*, 1972; Ichikawa and Fan, 1973) as well as branch and bound search methods (Lee *et al.*, 1970).

In terms of process flowsheets, the first computer-aided process synthesizer for generating initial structures, AIDES [Adaptive Initial DEsign Synthesizer], was developed by Rudd and his students (Siirola *et al.*, 1971; Siirola and Rudd, 1971; Powers, 1972); using a high level representation of tasks, it relied on the use of heuristics and linear programming, which were coordinated through a means–ends-analysis search. The second computer-aided process synthesizer to be developed was BALTAZAR (Mahalec and Motard, 1977a,b). It also relied on heuristics and linear programming, and used a tree search within the framework of theorem proving. Neither AIDES nor BALTAZAR incorporated equipment costs directly, but they employed heuristics as indicators of economic performance.

The current state of flowsheet synthesis is represented by two different approaches: (1) *hierarchical decomposition* (Douglas, 1985, 1988, 1990) and its computer implementation PIP [Process Invention Procedure], (Kirkwood *et al.*, 1988) and (2) *mathematical programming* (Grossmann, 1985, 1990a,b) and its initial computer implementation in PROSYN [PROcess SYNthesizer], (Kravanja and Grossmann, 1990). It has been pointed out (Rippin, 1990) that these two approaches are concerned with different aspects of design and can be regarded as complementary.

The hierarchical decomposition technique breaks the synthesis procedure into five discrete decision levels: (1) batch versus continuous, (2) input/output structure of flowsheet, (3) recycle structure and reactor considerations, (4) separation systems, and (5) heat exchanger network. At each decision level beyond the first, the economic potential of the project is evaluated and a decision is made whether or not further work on the project is justified. This method utilizes heuristics, short-cut design procedures, and physical insight to develop an initial base-case design. This approach is motivated by Douglas' claim that only 1% of all designs are implemented in practice; thus this screening procedure avoids detailed evaluation of most alternatives. Relying on heuristics, this approach cannot *rigorously* produce an optimal design; and although heuristics often lead to good designs, they are *fallible* (e.g., see Fonyó and Mizsey, 1990). Furthermore, owing to the *sequential* nature of the flowsheet synthesis, interactions among the design variables at the various decision levels may not be properly accounted for, as it is necessary to solve for them *simultaneously*. For instance, Duran and Grossmann (1986a) and later Lang *et al.* (1988) have shown that simultaneous optimization and heat integration of flowsheets generally produces significant improvements in the profit compared to the sequential approach. Despite these shortcomings, hierarchical decomposition provides a useful approach for generating an initial flowsheet and alternatives (i.e., a base-case design and superstructure). And, when coupled with the concept of simultaneous synthesis of the complete flowsheet, it also provides a framework for decomposing the synthesis problem into a hierarchy of detailed and aggregated models; it is therefore simpler to solve than the entire flowsheet, while still reflecting the presence of downstream tasks.

The mathematical programming approach utilizes optimization techniques to select the configuration and parameters of the processing system (Grossmann, 1985, 1990a,b). A superstructure containing alternative processing units and interconnections is modeled as discrete, binary variables (0–1) to depict the existence (1) or nonexistence (0) of that unit. Initially, the synthesis of total process systems (chemical plant, heat exchanger network, utility plant) was formulated as a mixed-integer linear program (MILP) (Papoulias and Grossmann, 1983a,b,c). To explicitly handle the nonlinearities in process models, the outer-approximation algorithm for MINLP has been developed and successively refined (Duran and Grossmann, 1986b; Kocis and Grossmann, 1987, Viswanathan and Grossmann, 1990). It is now widely available in the program DICOPT++ within the modeling system GAMS (Brooke *et al.*, 1988). This algorithm partitions the problem into two parts: (1) an NLP subproblem, where the continuous variables for a single flowsheet configuration are initially optimized and then the remaining alternative substructures are suboptimized for the given flows; (2) linearization of the nonlinear equations to obtain an MILP master problem, which then determines a new optimal flowsheet configuration (i.e., new set of binary variables) for the next NLP subproblem.

For convex MINLP problems (Duran and Grossmann, 1986b), the NLP optimization yields an upper bound for a cost-minimization problem while the MILP yields a lower bound. Thus, when the values of the objective functions for both problems are identical (or they cross), the global optimal solution is obtained. For the more general nonconvex MINLP problem (Viswanathan and Grossmann, 1990), the algorithm terminates when no improvement in the NLP objective function is obtained. In this case one cannot ascertain whether a local or the global optimal solution is obtained. This is because the linearizations of nonconvex equations may cut off portions of the feasible region, including the global optimal solution. Although the algorithm provides a reasonable degree of reliability for finding the global optimum, it can get trapped in a poor local solution (Daichendt and Grossmann, 1993a,b). The algorithmic approach presupposes that a superstructure is available or can be generated. At present, depending on the problem formulation, the number of binary variables needed to model the superstructure places an upper bound to the size of the problem that can be solved. It should also be noted that in order to address the MINLP optimization of process flowsheets more effectively, Kocis and Grossmann

(1989b) developed the modeling/decomposition (M/D) strategy in which the basic objective is to solve NLP subproblems pertaining only to the existing part of the superstructure. This strategy not only avoids the solution of NLP problems of larger dimensionality, but also reduces such numerical difficulties as singularities that arise in the case of nonexisting units with zero flows. The M/D strategy has been implemented in PROSYN and extended in various ways by Kravanja and Grossmann (1994).

B. Review of Subsystem Synthesis

Most of the research work in process synthesis has concentrated on the study of subsystems in which a major emphasis has been the management of energy. Pinch analysis has been widely used for setting design targets (see Linnhoff, 1993, for a recent review). However, recent efforts have been directed to the better understanding of synthesis issues in nonideal separation systems (Van Dongen and Doherty, 1985; Doherty and Cardarola, 1985) and reaction systems (Glasser *et al.,* 1987).

1. Heat-Exchanger Network Synthesis

Heat-exchanger network synthesis (HENS) is by far the most developed technique, and many methods and software packages are available for it. An extensive review can be found in Gundersen and Naess (1988). The discovery of the heat-recovery pinch (Umeda *et al.,* 1979; Linnhoff *et al.,* 1982), which is derived through thermodynamic analysis, provided the basis for the advancements in developing synthesis techniques for HENS. The most widely used method, commonly known as ''pinch technology'' (Linnhoff and Hindmarsh, 1983), relies on the use of targets (energy, number units, area) and is based on a user-driven approach. SUPERTARGET and ADVENT are two major pieces of software implementing this approach. As for algorithmic methods, there has been a gradual evolution from LP/MILP/NLP methods, which are based on targets (Cerda and Westerberg, 1983; Papoulias and Grossmann, 1983b; Floudas *et al.,* 1986; Colberg and Morari, 1990; Gundersen and Grossmann, 1990), to simultaneous MINLP models in which networks are automatically synthesized and energy, area, and number of units are optimized simultaneously (see Yee and Grossmann, 1990; Ciric and Floudas, 1991). Examples of software include MAGNETS for the target-based methods and SYNHEAT for the simultaneous MINLP. Also, the utility target for the HENS problem has the advantage that it can be effectively represented in an aggregated form (Duran and Grossmann, 1986a), and thus can be embedded within other synthesis models to perform simultaneous optimization. It should also be noted that the ideas of pinch analysis are being expanded beyond heat-exchanger networks to include total sites and assessment of environmental problems (see Linnhoff, 1993, for a review). Also, an important related problem is that of synthesis of utility systems (Papoulias and Grossmann, 1983a; Colmenares and Seider, 1987) and synthesis of integrated refrigeration systems (Shelton and Grossmann, 1986a,b).

2. *Distillation Sequencing*

A general review on distillation synthesis can be found in Westerberg (1985) and Floquet *et al.* (1988). The synthesis of simple separation sequences based on heuristics is fairly well developed. Enumeration search methods (dynamic programming, branch-and-bound) have been proposed by Hendry and Hughes (1972) and Gomez-Muñoz and Seader (1985), while evolutionary search procedures have been described by Stephanopoulos and Westerberg (1976), Seader and Westerberg (1977) and Nath and Motard (1981).

Thermodynamic analysis of the effect of composition on the work of separation has been used to suggest nonconventional ways (i.e., nonsharp splits) of separating mixtures by distillation (Petlyuk *et al.*, 1965). Further work in this area provides sequencing heuristics (Gomez-Muñoz and Seader, 1985). Graphical techniques (Carlberg and Westerberg, 1989a,b) have been developed that provide insight into the heat flows of these complex columns. An evolutionary synthesis strategy has been proposed (Koehler *et al.*, 1992) that decomposes the problem into a selection of sequences based on decision factors starting from an initial superstructure. Then the sequence is further refined by considering the introduction of side-stream strippers and enrichers.

Complete thermodynamic analysis, based on reversible distillation, takes into account the effects of finite temperature and composition driving forces as well as nonuniform heat distribution and hydraulic resistance (Fonyó, 1974a,b). The effect of nonuniform heat distribution (i.e., adiabatic distillation) can be mitigated by the introduction of intercoolers/interheaters (Terranova and Westerberg, 1989; Dhole and Linnhoff, 1992).

Algorithmic formulations for heat-integrated distillation sequence synthesis involving simple columns (Rathore *et al.*, 1974a,b; Andrecovich and Westerberg, 1985a,b; Kakhu and Flower, 1988; Floudas and Paules, 1988) have been developed as well as a graphical approach using the pinch design method (Smith and Linnhoff, 1988). A more accurate thermodynamic targeting model has recently been reported by Wahnschafft *et al.* (1993). In addition, work on superstructure optimization has been performed for synthesizing complex columns (Sargent and Gaminibandara, 1976; Eliceche and Sargent, 1986; Rigg, 1991). The synthesis of systems involving multiple feeds and mixed products has also been considered (Floudas, 1987; Wehe and Westerberg, 1987; Quesada and Grossmann, 1995a).

As for the synthesis of azeotropic columns, most of the developments have been based on geometric representations with residue curves (Doherty and Cardarola, 1985; Knight and Doherty, 1989; Van Dongen and Doherty, 1985). Guidelines have also been developed by Stichlmair *et al.* (1989), some of which Laroche *et al.* (1992) have questioned as general synthesis rules. Finally, Wahnschafft *et al.* (1992) have developed the program SPLIT, which is based on a ruled-based system and evolutionary search method.

3. Mass-Exchange Networks

Motivated by applications in waste recovery systems, El-Halwagi and Manousiouthakis (1989a,b, 1990) have considered the problem of synthesizing mass-exchange networks. For the simpler case in which concentration targets are specified for single components, interesting analogies can be drawn with the heat-exchanger network problem, both in terms of the pinch point (El-Halwagi and Manousiouthakis, 1989a) and in terms of the MILP transshipment model (El-Halwagi and Manousiouthakis, 1990). For the case of concentration specifications for multiple components, the synthesis problem becomes more complex and must be formulated as an MINLP problem (El-Halwagi and Manousiouthakis, 1989b; Papalexandri *et al.*, 1993). Bagajewicz and Manousiouthakis (1992) have extended the representation of mass-exchange networks for synthesizing heat-integrated distillation columns. These authors use a "state-space" approach for avoiding a simultaneous superstructure optimization, and they incorporate the constraints by Duran and Grossmann (1986a) for the pinch location method to represent the heat integration. A simultaneous MINLP model has been proposed recently by Papalexandri *et al.* (1993).

4. Reactor Network Synthesis

Synthesis of reactor networks poses a more difficult modeling problem as these must usually be described by differential-algebraic equations. Compared to HENS and distillation systems, however, the combinatorial part in reactor networks tends to be smaller. Heuristic-based approaches are generally limited to simple reactions. Reactor networks have been synthesized using a superstructure representation of serial recycle reactors without bypass for isothermal and nonisothermal reactors (Chitra and Govind, 1985a,b). A geometric approach, based on reactor trajectories, describes the attainable region of composition space generated by reaction and mixing for isothermal, constant-density systems (Glasser *et al.*, 1987), and was later extended to include adiabatic, variable-density systems (Hildebrandt *et al.*, 1990). An algorithmic procedure using a constant-dispersion model has been discussed (Achenie and Biegler, 1986). Subsequent work converted the synthesis problem into an optimal control problem

(Achenie and Biegler, 1988) and to an MINLP optimization problem (Kokossis and Floudas, 1991). Recent work has relied on targeting models (Balakrishna and Biegler, 1992a) in which a reactor network representation is based on a segregated mixing zone and multiple zones of maximum mixedness (multiple-compartment model), that taking advantage of the methods developed by Glasser *et al.* while overcoming the dimensional constraints by using two-dimensional projections of the multidimensional composition space. The simple segregated model is proposed as an initial solution, which can be posed as an LP and checked for global optimality. If it is not optimal, an iterative scheme is invoked that increases the complexity of the model by adding the first maximal-mixedness compartment, then checks to see if this more complex model can extend the attainable region. This method has also been extended to consider simultaneous heat integration (Balakrishna and Biegler, 1992b) using the aggregated model by Duran and Grossmann (1986a). Glavic *et al.* (1988) have developed thermodynamic criteria for the appropriate placement of reactors in a process flowsheet. Also, Omtveit and Lien (1993) have extended the representation of the attainable region so as to account for recycles in flowsheets.

III. Mathematical Programming Approach

From the review given in the previous section, it is clear that a significant number of synthesis models have been developed that are based on mathematical programming. It follows that the process synthesis problem can generally be stated as follows: Given specifications of input and of output streams—which may correspond to raw materials and desired products in flowsheets or simply to process streams in subsystems (e.g., heat-exchanger networks)—the problem consists in integrating a process system 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. One has to determine which units should integrate the system and how they should be interconnected, as well as the sizes and operating conditions of the units. The former clearly imply making discrete decisions, while the latter imply making a choice within a continuous space. Thus, from a conceptual standpoint the synthesis problem corresponds to a nonlinear 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. Postulate a representation of alternatives whose embedded designs are candidates for a feasible and optimal system.

Step 2. Model representation in step 1 as the MINLP problem

$$\begin{aligned} \min\ Z &= f(x,y) \\ \text{s.t.}\quad h(x,y) &= 0 \\ g(x,y) &\leq 0 \\ x \in X, &\qquad y \in Y \end{aligned} \qquad \text{(MINLP)}$$

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 material/heat flows, pressures, temperatures, and sizes of equipment.

Step 3. Obtain the optimal design embedded in the representation by solving the corresponding MINLP problem.

It is important to note that the representation of alternatives can be specified at various levels of detail. The two extreme cases are superstructures and aggregate models. Superstructures are relatively detailed representations of a process in which the potential existence of units and streams of a flowsheet are explicitly considered. Aggregate models are representations at higher levels of abstraction in which the synthesis problem is simplified by the use design targets (e.g. minimum utility, maximum yield). This may often simplify the mathematical programming problem, reducing the MINLP to a nonlinear programming (NLP), mixed-integer linear programming (MILP), or linear programming (LP) problem. Of course, several levels of abstraction are often possible for a given synthesis problem.

Umeda *et al.* (1972) were probably among the first authors to advocate the optimization of superstructures for process synthesis. At that time, however, the problem was formulated as a nonlinear programming problem involving only continuous variables, and solved with direct-search techniques. Next, Papoulias and Grossmann (1983c) formulated the simultaneous synthesis of total systems 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 algorithmic developments have allowed synthesis problems to be explicitly formulated as MINLP problems (Duran and Grossmann, 1986c; Kocis and Grossmann, 1989b; Viswanathan and Grossmann, 1988). As for the targeting approach, Cerda and Westerberg (1983) and Papoulias and Grossmann (1983b) were among the first to develop aggregated models for heat integration (LP transportation and LP transshipment models) with the objective of minimizing utility cost. Other targeting models include the ones proposed by El-Halwagi and Manousiouthakis (1989a) for mass-exchange networks, and the one by Balakrishna and Biegler (1992a) for reactor networks.

The two crucial steps in the approach just described are step 1, for generating the representation of alternatives, and step 3, for solving the MINLP problem. As it turns out, however, step 2 is also extremely important because the way in which one models MINLP problems can have a great impact on the performance of the algorithms. In the next sections, we will discuss the three major steps in the optimization methodology.

IV. Representation of Alternatives

In order to formulate the synthesis problem as an optimization problem, one has to develop a representation of alternatives that will systematically consider the candidates for the optimal solution. Developing an appropriate representation is clearly of paramount importance, as the optimal solution can be only as good as the representation being used.

We consider superstructure representations first. In general, these can be explicit or implicit. The former generally give rise to networks while the latter give rise to trees. As an example, consider the separation of a single feed of four 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 alternatives can be enumerated implicitly through a branch-and-bound search. However, in using this representation, the MINLP problem must be converted into the separable discrete optimization problem over the discrete space Y_{D_i}, $i = 1, ..., m$:

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

This requires that continuous variables be selected independently at each node of the tree, often leading to suboptimal solutions even if the branch-and-bound search is performed rigorously. Also, although fewer nodes may be examined in the tree with the use of heuristics, this increases the likelihood of obtaining suboptimal solutions.

On the other hand, consider the network representation shown in Fig. 2 that is implied by the MILP model of Andrecovich and Westerberg (1983). 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.,

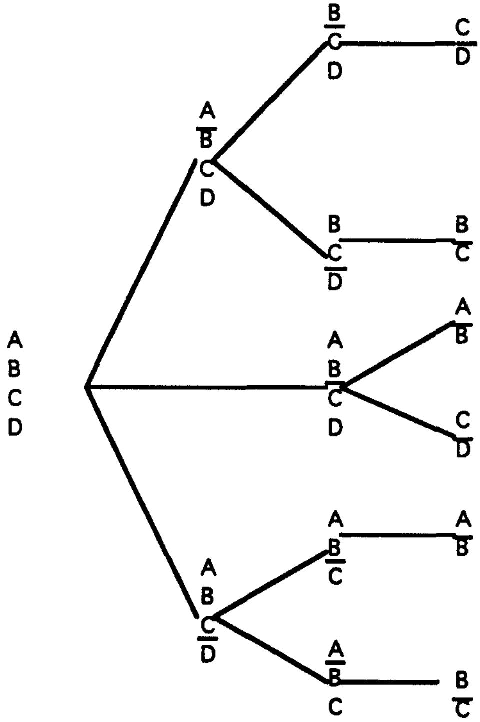


FIG. 1. Tree representation of separation sequence.

sequence 1 (A/BCD, B/CD, C/D) and sequence 2 (AB/CD, A/B, C/D) share the node C/D]. Using this network representation is more compact for modeling the problem explicitly as an MINLP problem. The advantage here is that the optimization can be performed rigorously because 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.

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

The next question we address is how to actually postulate or derive the superstructures. As experience—our own and that of other researchers—has shown, this is an easier task for homogeneous systems than for heterogeneous systems. An example is the network in Fig. 2, although it is restricted to sharp separations. We can, however, extend this representation to separation problems involving mixed products, with single feed columns that allow the possibility of bypasses and nonsharp splits as shown in Fig. 3 (see Floudas and Anastasiadis, 1988; Quesada and Grossmann, 1995a).

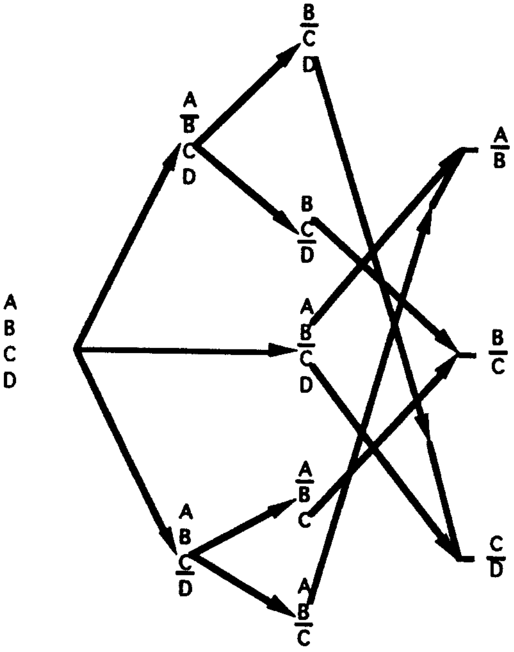


FIG. 2. Network representation of separation sequence.

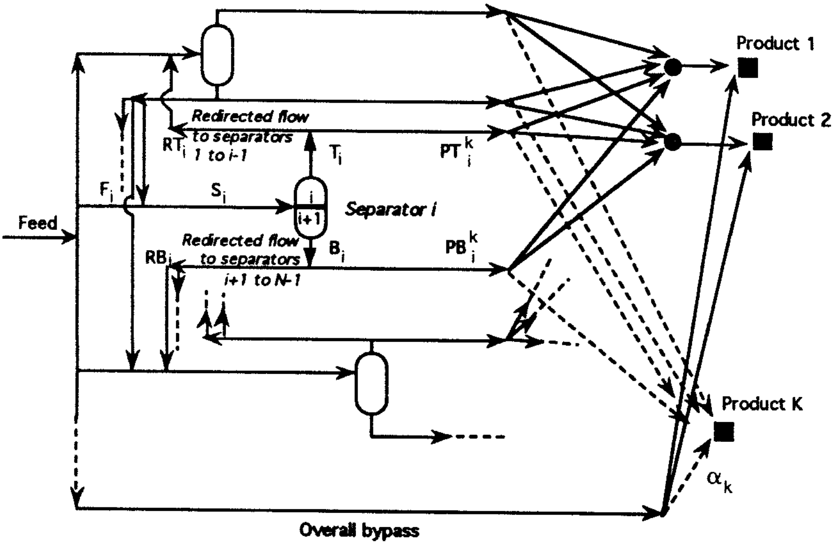


FIG. 3. Superstructure for single feed and multiple products.

As an additional example of an homogeneous system, consider the synthesis of heat-exchanger networks. Here, one possible representation that allows for systematic stream splitting and mixing is shown in Fig. 4. In this superstructure each exchanger unit corresponds to a potential match of pairs of streams. By activating the appropriate exchangers and setting flows to zero, all network configurations can be generated (see Floudas *et al.,* 1986; Ciric and Floudas, 1988). A different representation is given in Fig. 5. Here the superstructure for the network is represented by a sequence of stages where all potential stream matches can take place in each stage (Yee *et al.,* 1990a). Although this superstructure is more restricted, it has the advantage that, under assumptions of isothermal mixing, no flow variables are required in the MINLP and all the constraints become linear. Also, in this representation one can easily control the complexity of the network by imposing constraints to disallow the splitting of streams.

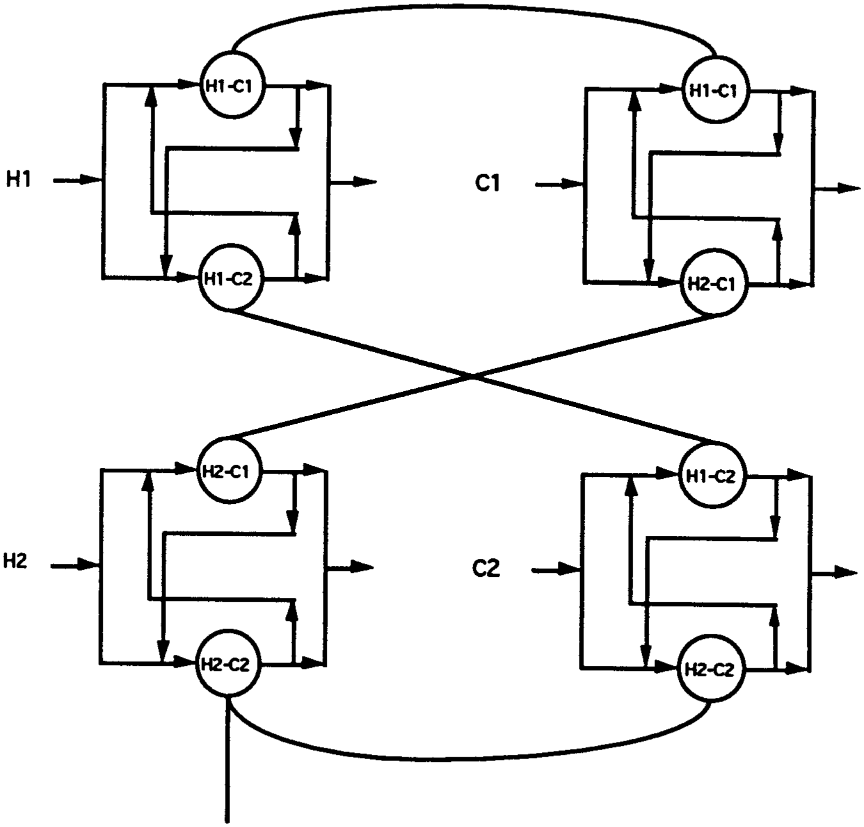


FIG. 4. Superstructure for heat-exchanger network proposed by Floudas *et al.* (1986).

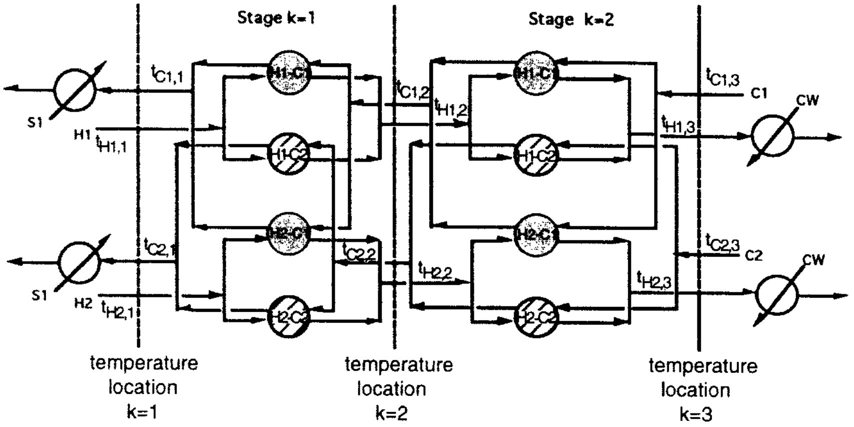


FIG. 5. Superstructure for heat exchanger networks proposed by Yee *et al.* (1990a).

In the previous examples, a one-to-one correspondence exists between the units and the tasks (e.g., in Fig. 2 each node performs a particular separation task). It is possible, however, to develop more general superstructure representations in which a one-to-many relationship exists between the units and the tasks. An example of a one-to-many relationship is the superstructure for separation shown in Fig. 6; proposed by Sargent and Gaminibandara (1976), this superstructure accommodates sharp splits and has the Petlyuk column embedded as an alternative design. Note, for instance, that column 1 does not have a prespecified separation task. From this example it is clear that superstructures that have one-to-many relationships between units and tasks tend to be richer in terms of embedded alternatives. On the other hand, the more restricted one-to-one superstructures tend to require simpler MINLP models that are quicker to solve.

Aggregated models, or higher-level representations of superstructures, can be developed to deal with simplified design objectives or targets. Here again, the example *par excellence* is the heat-exchanger network problem, where the targets for minimum utility cost and minimum number of units can be respectively modeled as LP and MILP transportation (Cerda and Westerberg, 1983) or transshipment problems (Papoulias and Grossmann, 1983b). Figure 7 shows the heat cascade representation used in the transshipment model for predicting the minimum utility cost. This representation, which is simply given in terms of heat flows across temperature intervals, has been shown by Daichendt and Grossmann (1993a) to be an aggregation of the superstructure of Fig. 5 by Yee *et al.* (1990a).

When the stream data are variable (flowrates and temperatures), the utility target problem can be modeled as a system of inequalities (see Duran and Grossmann, 1986a); and when only the flowrates are variables, it can be represented

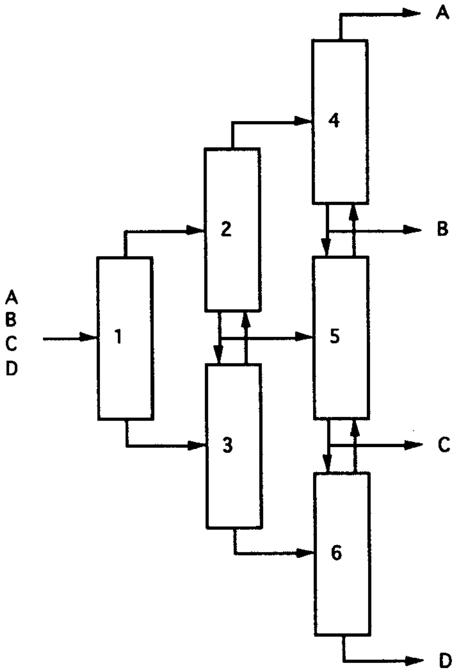


FIG. 6. Sargent–Gaminibandara superstructure.

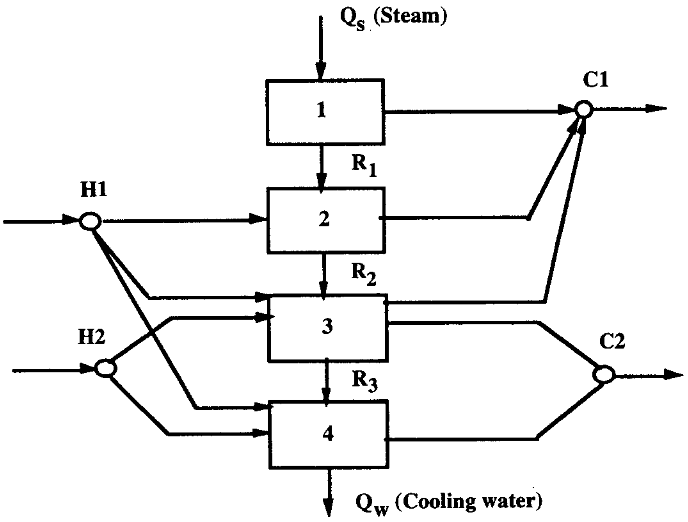


FIG. 7. Heat cascade of transshipment model proposed by Papoulias and Grossmann (1983a,b,c).

by the transportation or transhipment equations (see Papoulias and Grossmann, 1983c; Andrecovich and Westerberg, 1985b). But while these representations clearly simplify the synthesis problem, they do not provide all the information to automatically synthesize a network.

The systematic development of superstructures for heterogeneous systems is, in principle, a more difficult task. Consider, for instance a process flowsheet that is composed of reaction, separation, and heat integration subsystems. In theory, one could develop a superstructure by combining the superstructures for each subsystem. However, this approach could lead to a very large MINLP optimization problem.

Therefore, one approach is to assume that some preliminary screening has been performed (e.g., through heuristics) in order to postulate a smaller number of alternatives in the superstructure (Kocis and Grossmann, 1989b). While this approach seems restrictive, it does provide a systematic framework for analyzing specific alternatives at the level of tasks. As an example, consider the synthesis of an ammonia plant. A preliminary screening would indicate that the following major options: for the reactor, multibed quench or tubular; for separation of product, flash condensation or absorption/distillation; for recovery of hydrogen in purge, membrane separation or cryogenic separation. Figure 8 displays the superstructure for these alternatives. This superstructure has embedded at least eight different configurations.

As this example indicates, generating superstructures for heterogeneous systems based on specific alternatives at the level of tasks is actually not a very

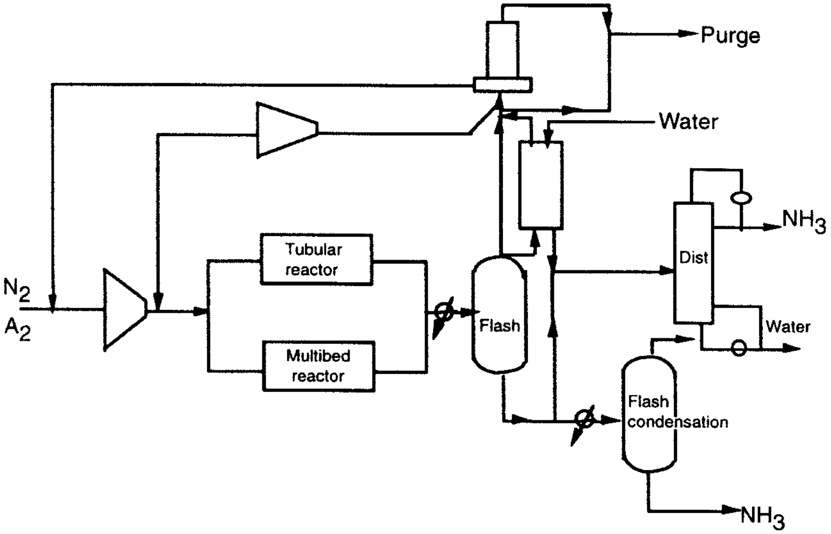


FIG. 8. Superstructure for ammonia process.

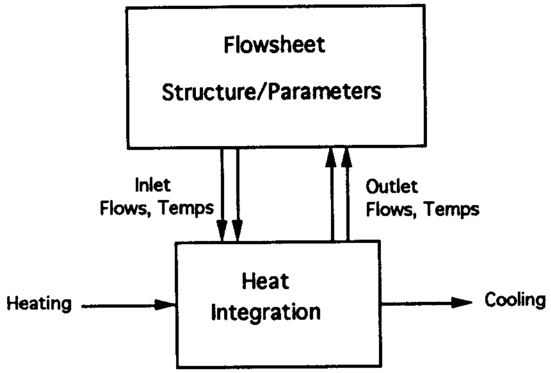


FIG. 9. Simultaneous optimization and heat integration.

difficult task. Another approach to developing synthesis models for heterogeneous systems is to use an aggregated representation for one of the subsystems in order to facilitate the simultaneous optimization of the entire system. For instance, this approach has been applied in the simultaneous optimization and heat integration of flowsheets in which utility loads, rather than a network structure, are predicted for the latter (see Fig. 9). In the work by Papoulias and Grossmann (1983c), the heat integration was represented through the transshipment model with variable flows, while in the work by Duran and Grossmann (1986a). Lang *et al.* (1988), and Balakrishna and Biegler (1992b), the heat integration is represented through a set of inequalities that are a function of variable flows and temperatures. Interestingly, the largest benefit of this approach is the fact that the simultaneous strategy yields designs exhibiting greater profit and overall conversion than those observed in sequential strategies. Sequential strategies tend to produce suboptimal solutions since interactions and trade-offs cannot be accounted for properly by designing each subsystem one after another.

Finally, it should be noted that one of the other approaches being examined for heterogeneous systems relies on physically based representations for reaction and mass and heat transfer, which are commonly treated as aggregated models. Also, Friedler *et al.* (1991, 1992, 1993) have provided graph-theoretic algorithms to systematically derive superstructures for process networks, given a list of unit operations and process streams.

V. MINLP Modeling

Once a representation for the candidate designs has been developed, the next step involves the modeling of the MINLP optimization problem. The major feature in such models is the modeling of discrete decisions, typically 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.

The handling of 0–1 variables allows the specification of constraints that 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:

$$\sum_{i \in I} y_i = 1 \tag{2}$$

Select at most one unit:

$$\sum_{i \in I} y_i \leq 1 \tag{3}$$

Select at least one unit:

$$\sum_{i \in I} y_i \geq 1 \tag{4}$$

(b) If–then conditions:

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

$$y_k - y_i \leq 0 \tag{5}$$

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:

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

which can be modeled through the constraint

$$Ly \leq x \leq Uy \tag{6}$$

This constraint is often used in conjunction with linear cost models with fixed cost charges, which again requires the use of 0–1 variables:

$$C = \alpha^{\mathrm{T}} y + \beta^{\mathrm{T}} x \tag{7}$$

$$Ly \leq x \leq Uy$$

While deriving integer constraints like those outlined above is not too difficult a task, deriving others can be more complex. The modeling of integer and mixed-integer constraints can, however, be facilitated by the use of propositional logic. It has been shown by a number of authors (e.g., Williams, 1988; Cavalier and Soyster, 1987) that virtually any propositional logic statement can be translated into a set of 0–1 inequalities by the systematic application of rules. For this, one must consider how basic logical operators can each be transformed into an equivalent representation in the form of an equation or inequality. These transformations are then used to convert general logical expressions into an equivalent mathematical representation (Cavalier and Soyster, 1987; Cavalier *et al.,* 1990).

Consider logic propositions that are composed of literals P_i which represent a given selection or action. To represent the truth of that literal, a binary variable y_i is assigned. The negation or complement of P_i ($\neg P_i$) is given by $1 - y_i$. The logical value of true corresponds to the binary value of 1 and false corresponds to the binary value of 0. The basic operators used in propositional logic and the representation of their relationships are shown in Table I. From this table, it is easy to verify, for instance, that the logical implication $P_1 \vee P_2$ reduces to the inequality in (4).

With the basic equivalent relations given in Table I (e.g., see Williams, 1988), one can systematically model an arbitrary propositional logic expression that is given in terms of OR, AND, IMPLICATION operators, as a set of linear equality and inequality constraints. One approach is to systematically convert the logical expression into its equivalent *conjunctive normal form* representation, which involves the application of pure logical operations (Raman and Grossmann, 1991). The conjunctive normal form is a conjunction of clauses, $Q_1 \wedge Q_2 \wedge \cdots \wedge Q_s$. Hence, for the conjunctive normal form to be true, each clause Q_i must be true independent of the others. Also, since a clause Q_i is just a disjunction of literals, $P_1 \vee P_2 \vee \cdots \vee P_r$, it can be expressed in the linear mathematical form as the inequality.

$$y_1 + y_2 + \cdots + y_r \geq 1 \tag{8}$$

The procedure to convert a logical expression into its corresponding conjunctive normal form was formalized by Clocksin and Mellish (1981). The systematic procedure consists of applying the following three steps to each logical proposition:

(1) Replace the implication by its equivalent disjunction,

$$P_1 \Rightarrow P_2 \quad \Leftrightarrow \quad \neg P_1 \vee P_2 \tag{9}$$

(2) Move the negation inward by applying DeMorgan's theorem:

$$\neg(P_1 \wedge P_2) \quad \Leftrightarrow \quad \neg P_1 \vee \neg P_2 \tag{10}$$

$$\neg(P_1 \vee P_2) \quad \Leftrightarrow \quad \neg P_1 \wedge \neg P_2 \tag{11}$$

(3) Recursively distribute the OR over the AND by using the following equivalence:

$$(P_1 \wedge P_2) \vee P_3 \quad \Leftrightarrow \quad (P_1 \vee P_3) \wedge (P_2 \vee P_3) \tag{12}$$

TABLE I

CONSTRAINT REPRESENTATION OF LOGIC PROPOSITIONS AND OPERATORS

Logical relation	Comments	Boolean expression	Representation as linear inequalities
Logical OR		$P_1 \vee P_2 \vee \cdots \vee P_r$	$y_1 + y_2 + \cdots + y_r \geq 1$
Logical AND		$P_1 \wedge P_2 \wedge \cdots \wedge P_r$	$y_1 \geq 1$ $y_2 \geq 1$ $\cdot\cdot$ $y_r \geq 1$
Implication	$P_1 \Rightarrow P_2$	$\neg P_1 \vee P_2$	$1 - y_1 + y_2 \geq 1$
Equivalence	P_1 if and only if P_2 $(P_1 \Rightarrow P_2) \wedge (P_2 \Rightarrow P_1)$	$(\neg P_1 \vee P_2) \wedge (\neg P_2 \vee P_1)$	$y_1 = y_2$
Exclusive OR	Exactly one of the propositions is true	$P_1 \underline{\vee} P_2 \underline{\vee} \cdots \underline{\vee} P_r$	$y_1 + y_2 + \cdots + y_r = 1$

Once each logical proposition has been converted into its conjunctive normal form representation, $Q_1 \wedge Q_2 \wedge \cdots \wedge Q_s$, it can be easily expressed as a set of linear equality and inequality constraints.

As an example, consider the logic condition "if the absorber is selected or the membrane separator is selected, then do not use cryogenic separation." Assigning the Boolean literals to each action P_{A} = select absorber, P_{M} = select membrane separator, P_{CS} = select cryogenic separation, the logic expression is given by

$$P_{\mathrm{A}} \vee P_{\mathrm{M}} \Rightarrow \neg \mathrm{P}_{\mathrm{CS}} \tag{13}$$

Removing the implication as in (9) yields

$$\neg(P_{\mathrm{A}} \vee P_{\mathrm{M}}) \vee \neg P_{\mathrm{CS}} \tag{14}$$

Applying De Morgan's Theorem as in (10) leads to

$$(\neg P_{\mathrm{A}} \wedge \neg P_{\mathrm{M}}) \vee \neg P_{\mathrm{CS}} \tag{15}$$

Distributing the AND over the OR gives

$$(\neg P_{\mathrm{A}} \vee \neg P_{\mathrm{CS}}) \wedge (\neg P_{\mathrm{M}} \vee \neg P_{\mathrm{CS}}) \tag{16}$$

Assigning the corresponding 0–1 variables to each term in the above conjunction and using (8), we get

$$\begin{aligned} 1 - y_{\mathrm{A}} + 1 - y_{\mathrm{CS}} &\geq 1 \\ 1 - y_{\mathrm{M}} + 1 - y_{\mathrm{CS}} &\geq 1 \end{aligned} \tag{17}$$

which can be rearranged to the two inequalities

$$\begin{aligned} y_{\mathrm{A}} + y_{\mathrm{CS}} &\leq 1 \\ y_{\mathrm{M}} + y_{\mathrm{CS}} &\leq 1 \end{aligned} \tag{18}$$

Finally, it is important to emphasize that the way one models an MINLP can have a great impact on the performance of the MINLP algorithm. A simple example is the logical constraint $x - Uy \leq 0$, where large values of U reduce the lower bound in the LP-based branch-and-bound search. This phenomenon has been widely recognized in the field of integer programming (e.g., see Williams, 1988; Nemhauser and Wolsey, 1988). The above procedure based on systematically deriving constraints from propositional logic will generally tend to produce well-posed constraints. For example, one logical constraint that may arise in a multiperiod problem states that installing a given unit ($z = 1$) implies that the unit can operate in any of the m time periods ($y_i = 1$, $i = 1, 2, \ldots, m$):

$$\sum_{i=1}^{m} y_i - mz \leq 0 \tag{19}$$

This constraint simply states that if $z = 0$, no operation of the unit is possible in the m time periods, while if $z = 1$, operation in any of the m periods is possible. While (19) is a "legitimate" constraint, it turns out that its equivalent representation by the set of inequalities

$$y_i - z \leq 0 \qquad i = 1, ..., m \tag{20}$$

which can be derived from the logic proposition "If unit operates in period 1, in period 2, ..., or in period m, then select unit," is a much more effective way to model the above logic condition. This is because its relaxation with continuous variables y_i yields extreme points, all with 0–1 values, which greatly reduces the computations in the branch-and-bound solution method.

The observations on modeling discussed above have led to the theoretical study in integer linear programming of facets of 0–1 polytopes for which algorithms are starting to emerge. These algorithms can systematically generate these types of constraints and reformulate a "badly" posed problem to improve its LP relaxation. (See, for example, Crowder *et al.*, 1983) for unstructured 0–1 linear problems, Van Roy and Wolsey (1987) and Balas *et al.* (1993) for MILP problems, and Schrijver (1986) for a review of the theoretical concepts.)

In MINLP, however, there is the additional complication that nonlinearities can often be formulated in many different, but equivalent, ways; and, as expected, this can have a great impact on the performance of MINLP algorithms, particularly with respect to the nonconvexities of the nonlinear constraints.

In general, the three major empirical guidelines for a "good" MINLP formulation are as follows:

1. Keep the problem as linear as possible.
2. Develop a formulation whose NLP relaxation is as tight 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 section on algorithms. It should be noted also that a new modeling and solution framework for discrete optimization is emerging that is based on disjunctive programming (Floudas and Grossmann, 1994). Some of these ideas will be covered later in the paper.

A. Example of a MINLP Model

To illustrate some of the modeling guidelines, we consider in this section the MINLP model for the synthesis of heat-exchanger networks by Yee and Grossmann (1990), which uses as a basis the superstructure in Fig. 5. The notation that will be used is as follows. Process streams are divided into two sets, set

HP for hot streams represented by index i and set CP for cold streams represented by index j. Index k is used to denote the superstructure stages given by the set ST. Indices HU and CU correspond to the heating and cooling utilities, respectively. Also, the following parameters and variables are used in the formulation:

Parameters:

TIN = inlet temperature of stream
F = heat capacity flow rate
CCU = unit cost for cold utility
CF = fixed charge for exchangers
β = exponent for area cost
Ω = upper bound for heat exchange

TOUT = outlet temperature of stream
U = overall heat transfer coefficient
CHU = unit cost of hot utility
C = area cost coefficient
NOK = total number of stages
Γ = upper bound for temperature difference

Variables:

dt_{ijk} = temperature approach for match (i,j) at temperature location k
$d t\mathrm{cu}_i$ = temperature approach for the match of hot stream i and cold utility
$d t\mathrm{hu}_j$ = temperature approach for the match of cold stream j and hot utility
q_{ijk} = heat exchanged between hot process stream i and cold process stream j in stage k
$q\mathrm{cu}_i$ = heat exchanged between hot stream i and cold utility
$q\mathrm{hu}_j$ = heat exchanged between hot utility and cold stream j
$t_{i,k}$ = temperature of hot stream i at hot end of stage k
$t_{j,k}$ = temperature of cold stream j at hot end of stage k
z_{ijk} = binary variable to denote existence of match (i,j) in stage k
$z\mathrm{cu}_i$ = binary variable to denote that cold utility exchanges heat with stream i
$z\mathrm{hu}_j$ = binary variable to denote that hot utility exchanges heat with stream j

With the above definitions, the formulation can now be presented.

(a) *Overall heat balance for each stream:* An overall heat balance is needed to ensure sufficient heating or cooling of each process stream. The constraints specify that the overall heat transfer requirement of each stream must equal the sum of the heat it exchanges with the other process streams at each stage plus the exchange with the utility streams:

$$\begin{aligned} (\mathrm{TIN}_i - \mathrm{TOUT}_i)F_i &= \sum_{k \in \mathrm{ST}} \sum_{j \in \mathrm{CP}} q_{ijk} + q\mathrm{cu}_i \qquad i \in \mathrm{HP} \\ (\mathrm{TOUT}_j - \mathrm{TIN}_j)F_j &= \sum_{k \in \mathrm{ST}} \sum_{i \in \mathrm{HP}} q_{ijk} + q\mathrm{hu}_j \qquad j \in \mathrm{CP} \end{aligned} \tag{21}$$

(b) *Heat balance at each stage:* An energy balance is also needed at each stage of the superstructure to determine the temperatures. Note that for the two-stage superstructure shown in Fig. 5, three temperatures, t_k, are required. Temperatures for the streams are highest at temperature location $k = 1$ and lowest at $k = 3$.

By assuming isothermal mixing, no variables are required for the flows, thus avoiding the introduction of bilinear constraints for the heat balances:

$$\begin{aligned} (t_{i,k} - t_{i,k+1})F_i &= \sum_{j\in CP} q_{ijk} \qquad k \in ST, \quad i \in HP \\ (t_{j,k} - t_{j,k+1})F_j &= \sum_{i\in HP} q_{ijk} \qquad k \in ST, \quad j \in CP \end{aligned} \tag{22}$$

(c) *Assignment of superstructure inlet temperatures:* Fixed supply temperatures (TIN) of the process streams are assigned as the inlet temperatures to the superstructure. In Fig. 5, for hot streams, the superstructure inlet corresponds to temperature location $k = 1$, while for cold streams, the inlet corresponds to location $k = 3$.

$$\begin{aligned} \mathrm{TIN}_i &= t_{i,1} \\ \mathrm{TIN}_j &= t_{j,\mathrm{NOK}+1} \end{aligned} \tag{23}$$

(d) *Feasibility of temperatures:* Constraints are also needed to specify a monotonic decrease of temperature at each successive stage. In addition, a bound is set for the outlet temperatures of the superstructure at the respective stream's target temperature. Note that the outlet temperature of each stream at its last stage does not necessarily correspond to the stream's target temperature since utility exchanges can occur at the outlet of the superstructure.

$$\begin{aligned} t_{i,k} &\geq t_{i,k+1} \qquad k \in ST, \quad i \in HP \\ t_{j,k} &\geq t_{j,k+1} \qquad k \in ST, \quad j \in CP \\ \mathrm{TOUT}_i &\leq t_{i,\mathrm{NOK}+1} \qquad i \in HP \\ \mathrm{TOUT}_j &\geq t_{j,1} \qquad j \in CP \end{aligned} \tag{24}$$

(e) *Hot and cold utility load:* Hot and cold utility requirements are determined for each process stream in terms of the outlet temperature in the last stage and the target temperature for that stream. The utility heat load requirements are determined as follows:

$$\begin{aligned} (t_{i,\mathrm{NOK}+1} - \mathrm{TOUT}_i)F_i &= q\mathrm{cu}_i \qquad i \in HP \\ (\mathrm{TOUT}_j - t_{j,1})F_j &= q\mathrm{hu}_j \qquad j \in CP \end{aligned} \tag{25}$$

(f) *Logical constraints:* Logical constraints and binary variables are needed to determine the existence of process match (i,j) in stage k and also any match involving utility streams. The 0–1 binary variables are represented by z_{ijk} for process stream matches, $z\mathrm{cu}_i$ for matches involving cold utility and $z\mathrm{hu}_j$ for matches involving hot utility. An integer value of 1 for any binary variable designates that the match is present in the optimal network. The constraints then are as follows:

$$\begin{aligned}
&q_{ijk} - \Omega z_{ijk} \leq 0 && \mathrm{i} \in \mathrm{HP},\ j \in \mathrm{CP},\ k \in \mathrm{ST} \\
&q\mathrm{cu}_i - \Omega z\mathrm{cu}_i \leq 0 && i \in \mathrm{HP} \\
&q\mathrm{hu}_j - \Omega z\mathrm{hu}_j \leq 0 && j \in \mathrm{CP} \\
&z_{ijk},\ z\mathrm{cu}_i,\ z\mathrm{hu}_j = 0,1
\end{aligned} \tag{26}$$

(g) *Calculation of approach temperatures:* The area requirement of each match will be incorporated in the objective function. Calculation of these areas requires that approach temperatures be determined. To ensure feasible driving forces for the exchangers selected in the optimization procedure, the binary variables are used to activate or deactivate the following constraints for approach temperatures:

$$\begin{aligned}
dt_{ijk} &\leq t_{i,k} - t_{j,k} + \Gamma(1 - z_{ijk}) && k \in \mathrm{ST},\ i \in \mathrm{HP},\ j \in \mathrm{CP} \\
dt_{ijk+1} &\leq t_{i,k+1} - t_{j,k+1} + \Gamma(1 - z_{ijk}) && k \in \mathrm{ST},\ i \in \mathrm{HP},\ j \in \mathrm{CP} \\
dt\mathrm{cu}_i &\leq t_{i,\mathrm{NOK}+1} - \mathrm{TOUT}_{\mathrm{CU}} + \Gamma(1 - z\mathrm{cu}_i) && i \in \mathrm{HP} \\
dt\mathrm{hu}_i &\leq \mathrm{TOUT}_{\mathrm{HU}} - t_{j,1} + \Gamma(1 - z\mathrm{hu}_j) && j \in \mathrm{CP}
\end{aligned} \tag{27}$$

Note that these constraints can be expressed as inequalities because the cost of the exchangers decreases with higher values for the temperature approaches *dt*. Also, the role of the binary variables in the constraints is to ensure that nonnegative driving forces exist for a selected match. When a match *(i,j)* occurs in stage *k*, $z_{ijk} = 1$ and the constraint becomes active so that the approach temperature is properly calculated. However, when the match does not occur, $z_{ijk} = 0$, and the contribution of the upper bound Γ on the right-hand side renders the constraint inactive. Note that the upper bounds can be set to zero for the utility exchangers since, for the data given, all the temperature differences are always positive. Also, one can specify a minimum approach temperature so that in the network, the temperature between the hot and cold streams at any point of any exchanger will be at least TMAPP:

$$dt_{ijk} \geq \mathrm{TMAPP} \tag{28}$$

(h) *Objective function:* Finally, the objective function can be defined as the annual cost for the network. The annual cost involves the combination of the utility cost, the fixed charges for the exchangers, and the area cost for each exchanger. LMTD, which is the driving force for a countercurrent heat exchanger, is approximated using the Chen approximation (1987):

$$\mathrm{LMTD} \sim [(dt1 * dt2) * (dt1 + dt2)/2]^{1/3}$$

This approximation is used to avoid the numerical difficulties of the LMTD equation when the approach temperatures for both sides of the exchanger are equal. Furthermore, when the driving force on either side of the exchanger equals zero, the driving force will be approximated to zero. The objective function is defined as follows:

$$\begin{aligned}
\min \sum_{i\in \mathrm{HP}} & \mathrm{CCU}\ q\mathrm{cu}_i + \sum_{j\in \mathrm{CP}} \mathrm{CHU}\ q\mathrm{hu}_j \\
& + \sum_{i\in \mathrm{HP}} \sum_{j\in \mathrm{CP}} \sum_{k\in \mathrm{ST}} \mathrm{CF}_{ij} z_{ijk} + \sum_{i\in \mathrm{HP}} \mathrm{CF}_{i,\mathrm{CU}} z\mathrm{cu}_i + \sum_{j\in \mathrm{CP}} \mathrm{CF}_{j,\mathrm{HU}} z\mathrm{hu}_j \\
& + \sum_{i\in \mathrm{HP}} \sum_{j\in \mathrm{CP}} \sum_{k\in \mathrm{ST}} \mathrm{C}_{ij}[q_{ijk}/(U_{ij}[(dt_{ijk}\ dt_{ijk+1})(dt_{ijk} + dt_{ijk+1})/2]^{1/3})]^{\beta_{ik}} \\
& + \sum_{i\in \mathrm{HP}} \mathrm{C}_{i,\mathrm{CU}}[q\mathrm{cu}_i/(U_{i,\mathrm{CU}}[(dt\mathrm{cu}_i)(\mathrm{TOUT}_i - \mathrm{TIN}_{\mathrm{CU}}) \qquad (29) \\
& \quad \{dt\mathrm{cu}_i + (\mathrm{TOUT}_i - \mathrm{TIN}_{\mathrm{CU}})\}/2]^{1/3})]^{\beta_{i,\mathrm{CU}}} \\
& + \sum_{j\in \mathrm{CP}} C_{\mathrm{HU},j}[q\mathrm{hu}_j/(U_{\mathrm{HU},j}[(dt\mathrm{hu}_j)(\mathrm{TIN}_{\mathrm{HU}} - \mathrm{TOUT}_j) \\
& \quad \{dt\mathrm{hu}_j + (\mathrm{TIN}_{\mathrm{HU}} - \mathrm{TOUT}_j)\}/2]^{1/3})]^{\beta_{j\mathrm{HU}}}
\end{aligned}$$

where

$$\frac{1}{U_{ij}} = \frac{1}{h_i} + \frac{1}{h_j}$$

The MINLP model for the synthesis problem consists of minimizing the objective function in (29) subject to the feasible space defined by eqs. (21)–(28). The continuous variables (t, q, $q$hu, qcu, dt, dtcu, dthu) are non-negative and the discrete variables z, $z$cu, $z$hu are 0–1. The advantage of this model is that the constraints (21)–(28) are all linear. The nonlinearities have all been placed in the objective function (29). However, it should be noted that since these terms are nonconvex, the MINLP may lead to local optimal solutions.

It should be noted that the simplifying assumption of isothermal mixing at the stage outlets for the stream splits is rigorous for the case when the network to be synthesized does not involve stream splits. For structures where splits are present, the assumption may lead to an overestimation of the area cost since it will restrict trade-offs of area between the exchangers involved with the split streams. In this case, one possibility is to refine the temperatures by introducing flow variables in the selected network structure and perform the corresponding optimization through an NLP model similar to the one given above.

Finally, an interesting feature of the MINLP model is that it is possible to add constraints to avoid generating structures with stream splits. This is accomplished simply by requiring that not more than one match be selected for every stream at each stage; that is,

$$\begin{aligned} &\sum_{i \in \mathrm{HP}} z_{ijk} \leq 1 \qquad j \in \mathrm{CP},\ k \in \mathrm{ST} \\ &\sum_{j \in \mathrm{CP}} z_{ijk} \leq 1 \qquad i \in \mathrm{HP},\ k \in \mathrm{ST} \end{aligned} \tag{30}$$

VI. MINLP Algorithms

A. Basic Algorithms

While there is a vast body of literature on LP, NLP, and integer LP with special structures, MINLP has received much less attention. In part, the explanation for this is that MINLP has been traditionally regarded as a very difficult problem because it is an NP-hard problem (Nemhauser and Wolsey, 1988) that is prone to combinatorial explosion. In our view, however, it is a mistake to regard these problems as ''unsolvable.'' The applications for MINLP are extremely rich; moreover, with current methods and technology, one can already solve problems of significant size and complexity. Furthermore, 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. Several examples of successful solution to optimality of integer programs of very large size can be found, for instance, in Crowder *et al.* (1983), Van Roy and Wolsey (1987), and Balas *et al.* (1993).

Our objective in this section is to provide a general overview of the basic MINLP algorithms, emphasizing their fundamental ideas and properties.

The most basic form of an MINLP problem when represented in algebraic form is as follows (equations are temporarily excluded):

$$\begin{aligned} &\min Z = f(x,y) \\ &\text{s.t. } g_j(x,y) \leq 0 \qquad j \in J \\ &x \in X, \qquad y \in Y \end{aligned} \tag{P1}$$

where $f(\cdot)$, $g(\cdot)$ are convex, differentiable functions, and x and y are the continuous and discrete variables, respectively. The set X is commonly assumed to be a compact set, e.g. $X = \{x \mid x \in \mathbf{R}^n,\ Dx \leq d,\ x^{\mathrm{L}} \leq x \leq x^{\mathrm{U}}\}$; the discrete set Y

is a polyhedral set of integer points, $Y = \{y \mid y \in \mathbf{Z}^m, Ay \leq a\}$, which in most applications is restricted to 0–1 values, $y \in \{0,1\}^m$. In most applications of interest the objective and constraint functions $f(\cdot)$, $g(\cdot)$ are linear in y (e.g., fixed cost charges and logic constraints).

Methods that have addressed the solution of problem (P_1) include the branch-and-bound method (BB) (Gupta and Ravindran, 1985; Nabar and Schrage, 1991; Borchers and Mitchell, 1992), the generalized Benders decomposition (GBD) method (Geoffrion, 1972), the outer-approximation (OA) method (Duran and Grossmann, 1986c; Yuan *et al.*, 1988; Fletcher and Leyffer, 1994), the LP/NLP-based branch-and-bound method (Quesada and Grossmann, 1992), and the extended cutting-plane (ECP) method (Westerlund and Pettersson, 1992).

There are three basic NLP subproblems that can be considered for problem (P_1).

(a) NLP relaxation:

$$\begin{array}{ll} \min Z_{\mathrm{LB}}^k = f(x,y) & \\ \text{s.t. } g_j(x,y) \leq 0 & j \in J \\ x \in X, & y \in Y_R \\ y_i \leq \alpha_i^k & i \in I_{\mathrm{FL}}^k \\ y_i \leq \beta_i^k & i \in I_{\mathrm{FU}}^k \end{array} \qquad \text{(NLP1)}$$

where Y_R is the continuous relaxation of the set Y, and I_{FL}^k, I_{FU}^k are index subsets of the integer variables y_i, $i \in I$, which are restricted to lower and upper bounds, α_i^k, β_i^k, at the kth step of a branch-and-bound enumeration procedure. Note that $\alpha_i^k = \lfloor y_i^l \rfloor$, $\beta_i^k = \lceil y_i^m \rceil$, $l < k$, $m < k$, where are y_i^l, y_i^m noninteger values at a previous step.

Also note that if $I_{\mathrm{FU}}^k = I_{\mathrm{FL}}^k = \emptyset$, $k = 0$, then (NLP1) corresponds to the continuous NLP relaxation of (P1); otherwise, it corresponds to the kth step in a branch-and-bound search. Also, the optimal objective function Z_{LB}^0 provides an absolute lower bound to (P1); for $m \geq k$, the bound is only valid for $I_{\mathrm{FL}}^k \subseteq I_{\mathrm{FL}}^m$, $I_{\mathrm{FU}}^k \subseteq I_{\mathrm{FL}}^m$.

(b) NLP subproblem for fixed y^k:

$$\begin{array}{ll} \min Z_{\mathrm{U}}^k = f(x,y^k) & \\ \text{s.t. } g_j(x,y^k) \leq 0 & j \in J \\ x \in X & \end{array} \qquad \text{(NLP2)}$$

which clearly yields an upper bound Z_{U}^k to (P1), provided (NLP2) has a feasible solution (which may not always be the case).

(c) Feasibility subproblem for fixed y^k.

$$\min_{x \in X} \left[\sum_{j \in J} \max\,[0,\ \{g_j(x,\ y^k)\}^p] \right]^{1/p} \qquad \text{(NLPF)}$$

which for the 1-norm (p = 1) leads to

$$\begin{aligned} &\min \sum_{j \in J} s_j \\ &\text{s.t. } g_j(x, y^k) \le s_j \qquad j \in J \\ &x \in X, \qquad s_j \in R^1, \qquad j \in J \end{aligned} \qquad \text{(NLPF-1)}$$

where s_j are slack variables.

For the infinity-norm ($p = \infty$) problem (NLPF) yields,

$$\begin{aligned} &\min u \\ &\text{s.t. } g_j(x, y^k) \le u \qquad j \in J \\ &x \in X, \qquad u \in R^1 \end{aligned} \qquad \text{(NLPF-}\infty\text{)}$$

The new predicted values y^K (or $(y^K,\ x^K)$) are obtained from a cutting-plane MILP problem that is based on the K points, $(x^k,\ y^k)$, k = 1, ..., K generated at the K previous steps:

$$\min Z_{\mathrm{L}}^K = \alpha \qquad \text{(M-MIP)}$$

$$\left.\begin{aligned} &\text{s.t. } \alpha \ge f(x^k, y^k) + \nabla f(x^k, y^k)^{\mathrm{T}} \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \\ &g_j(x^k, y^k) + \nabla g_j(x^k, y^k)^{\mathrm{T}} \begin{bmatrix} x - x^k \\ y - y^k \end{bmatrix} \le 0 \qquad j \in J^k \end{aligned}\right\} k = 1, \ldots, K$$

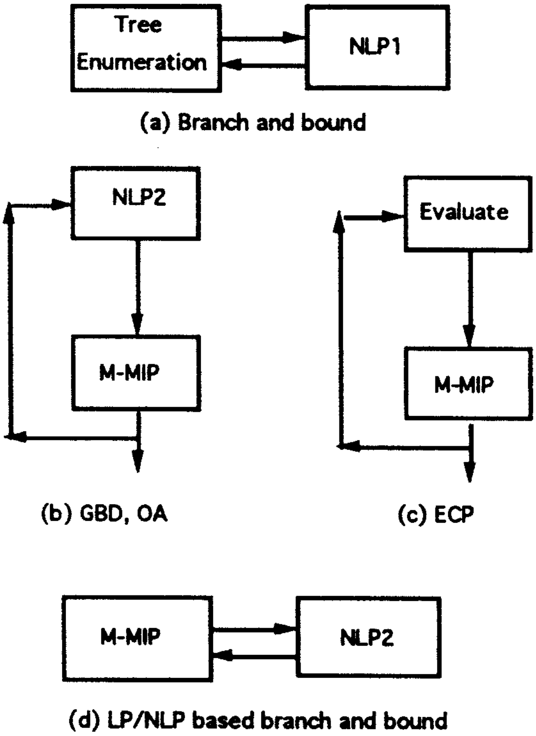
$$x \in X, \quad y \in Y, \quad \alpha \in R^1$$

where Z_{L}^K yields a valid lower bound to problem (P1). This bound is nondecreasing with the number of linearization points K.

The different methods can be classified according to the use of the subproblems (NLP1), (NLP2), and (NLPF) and the specific specialization of the MILP (M-MIP) (see Fig. 10). Note that in cases (b) and (d), (NLPF) is used if infeasible subproblems are found.

1. Branch and Bound

The BB method (Gupta and Ravindran, 1985; Nabar and Schrage, 1991; Borchers and Mitchell, 1992) starts by solving first the continuous NLP relaxation. If all discrete variables take discrete values, the search is stopped. Otherwise, it performs a tree search in the space of the integer variables y_i, $i \in I$,



(a) Branch and bound

(b) GBD, OA

(c) ECP

(d) LP/NLP based branch and bound

FIG. 10. Major steps in the different algorithms.

and solves a sequence of relaxed NLP subproblems of the form (NLP1) which yield lower bounds; fathoming of nodes occurs when the lower bound exceeds the current upper bound, or when all integer variables y_i take on discrete values. This method is attractive only if the NLP subproblems are relatively inexpensive to solve, or only few of them need to be solved. This could happen either when the dimensionality of the discrete variables is low or when the continuous NLP relaxation of (P1) is tight.

2. *Outer Approximation*

The OA method (Duran and Grossmann, 1986c; Yuan *et al.*, 1988; Fletcher and Leyffer, 1994) arises when NLP subproblems (NLP2) and MILP master problems (M-MIP) with $J^k = J$ are solved successively in a cycle of iterations to generate the points (x^k, y^k). The (NLP2) subproblems yield an upper bound that corresponds to the best current solution, $\mathrm{UB}^K = \min\ (Z_U^k)$. The master problems (M-MIP) yield a non-decreasing sequence of lower bounds $(Z_L^K$ since linearizations are accumulated as seen in (M-MIP). The cycle of iterations is stopped when the lower and upper bounds are within a specified tolerance. Also, if infeasible NLP subproblems are found, the feasibility problem (NLPF) is solved to provide the point x^k (commonly NLPF$-\infty$). The OA method generally requires relatively few cycles or major iterations. It trivially converges in one iteration if $f(x,y)$ and $g(x,y)$ are linear. It is also important to note that the MILP master problem need not be solved to optimality. In fact, given the upper bound UB^K and a tolerance ϵ, it is sufficient to generate the new $(y^K\ x^K)$ by solving

$$\begin{aligned}
&\min Z^K = 0 \cdot \alpha \qquad\qquad \text{(M-MIPF)}\\
&\text{s.t. } \alpha \le \mathrm{UB} - \epsilon\\
&\left.\begin{aligned}
&\alpha \ge f(x^k,y^k) + \nabla f(x^k,y^k)^{\mathrm{T}} \begin{bmatrix} x-x^k \\ y-y^k \end{bmatrix}\\
&g(x^k,y^k) + \nabla g(x^k,y^k)^{\mathrm{T}} \begin{bmatrix} x-x^k \\ y-y^k \end{bmatrix} \le 0
\end{aligned}\right\} k = 1, \ldots, K\\
&x \in X, \qquad y \in Y, \qquad \alpha \in R^1
\end{aligned}$$

3. Generalized Benders Decomposition

The GBD method (Geoffrion, 1972) is similar to the OA method. The difference arises in the definition of the MILP master problem (M-MIP). In the GBD method, only active inequalities are considered $J^k = \{j/g_j\ (x^k,\ y^k) = 0\}$ and the set $x \in X$ is disregarded. As shown in Quesada and Grossmann (1992), (M-MIP) reduces to a problem projected in the y-space:

$$\begin{aligned}
&\min Z_L^K = \alpha\\
&\text{s.t. } \alpha \ge f(x^k,y^k) + \nabla_y f(x^k,y^k)^{\mathrm{T}}(y-y^k) \qquad\qquad \text{(M-GBD)}\\
&\qquad + (\mu^k)^{\mathrm{T}}[g(x^k,y^k) + \nabla_y g(x^k,y^k)^{\mathrm{T}}(y-y^k)] \qquad k \in \mathrm{KFS}\\
&(\lambda^k)^{\mathrm{T}}[g(x^k,y^k) + \nabla_y g(x^k,y^k)^{\mathrm{T}}(y-y^k)] \le 0 \qquad k \in \mathrm{KIS}\\
&y \in Y, \qquad \alpha \in R^1
\end{aligned}$$

where KFS is the set of feasible subproblems (NLP2) and KIS is the set of infeasible subproblems whose solution is given by (NLPF$-\infty$). Also, $|\ \mathrm{KFS} \cup \mathrm{KIS}\ | = K$. As has been shown by Duran and Grossmann (1986c), the lower bounds of the OA method are greater than or equal to those of the GBD method. For this reason, GBD commonly requires a larger number of cycles or major iterations. As the number of 0–1 variables increases, this difference becomes more pronounced; therefore user-supplied constraints must be added to the master problem to strengthen the bounds (Sahinidis and Grossmann, 1991a). As has been shown by Sahinidis and Grossmann (1991b), GBD converges in one iteration once the optimal (x^*, y^*) has been found when problem (P1) has zero integrality gap. Also, Turkay and Grossmann (1994) have proved that performing one Benders iteration on the MILP master of OA is equivalent to a GBD iteration.

4. *Extended Cutting Plane*

The ECP method (Westerlund and Pettersson, 1992), which is an extension of Kelly's cutting plane algorithm for convex NLP (Kelley, 1960), does not rely on the use of NLP subproblems and algorithms. It relies only on the iterative solution of the problem (M-MIP) by successively adding the most violated constraint at the predicted point (x^k, y^k): $J^k = \{\hat{j} | \hat{j} \in \arg \{\max_{j \in J} g_j\,(x^k, y^k)\,\}\}$. Convergence is achieved when the maximum constraint violation lies within the specified tolerance. The optimal objective value of (M-MIP) yields a nondecreasing sequence of lower bounds. Note that since the discrete and continuous variables are converged simultaneously, a large number of iterations may be required. Also, the objective must be defined as a linear function.

5. *LP/NLP-Based Branch and Bound*

This method (Quesada and Grossmann, 1992) avoids the complete solution of (M-MIP) at each major iteration. The basic idea consists of performing an LP-based BB method for (M-MIP), solving NLP subproblems (NLP1) at those nodes in which feasible integer solutions are found. By updating the representation of the master problem in the open nodes of the tree with the addition of the corresponding linearizations, one eliminates the need to restart the tree search. This method can be applied both to OA and GBD methods, and commonly reduces quite significantly the number of nodes enumerated. The tradeoff, however, is that the number of NLP subproblems may increase. Computational experience has indicated that the number of NLP subproblems often remains unchanged. Also, Leyffer (1993) has reported substantial savings with this method.

B. Extensions of MINLP Methods

In this section, we present an overview of some of the major extensions of the methods presented in the previous section.

1. Quadratic Master Problems

For most problems of interest, problem (P1) is linear in y: $f(x,y) = \phi(x) + c^{\mathrm{T}}y$, $g(x,y) = h(x) + By$. When this is not the case Fletcher and Leyffer (1994) suggest including a quadratic approximation to (M-MIPF) of the form

$$
\begin{aligned}
\min Z^K = \alpha + \frac{1}{2}\begin{pmatrix} x-x^K \\ y-y^K \end{pmatrix}^{\mathrm{T}} \nabla^2\mathcal{L}(x^K,y^K)\begin{pmatrix} x-x^k \\ y-y^k \end{pmatrix} \\
\text{s.t. } \alpha \le \mathrm{UB}^K - \epsilon \qquad\qquad \text{(M-MIQP)} \\
\left.\begin{aligned}
\alpha \ge f(x^k,y^k) + \nabla f(x^k,y^k)^{\mathrm{T}}\begin{bmatrix} x-x^k \\ y-y^k \end{bmatrix} \\
g(x^k,y^k) + \nabla g(x^k,y^k)^{\mathrm{T}}\begin{bmatrix} x-x^k \\ y-y^k \end{bmatrix} \le 0
\end{aligned}\right\} \quad k = 1, \ldots, K \\
x \in X, \qquad y \in Y, \qquad \alpha \in R^1
\end{aligned}
$$

where $\nabla^2\mathcal{L}(x^K, y^K)$ is the Hessian of the Lagrangian of the last NLP subproblem. Note that Z^K does not predict valid lower bounds in this case. As noted by Ding and Sargent (1992), who developed a master problem similar to M-MIQP, the quadratic approximations can help to reduce the number of major iterations since an improved representation of the continuous space is obtained. This, however, comes at the price of having to solve an MIQP instead of an MILP. Note also that using (M-MIQP) for convex $f(x,y)$ and $g(x,y)$ leads to rigorous solutions since the outer approximations remain valid.

2. Reducing the Dimensionality of the Master Problem in OA

The master problem (M-MIP) can be rather large in the OA method. One option is to keep only the last linearization point, but this may lead to nonconvergence even in convex problems (e.g., see Bremicker *et al.*, 1990). A rigorous reduction of dimensionality without greatly sacrificing the strength of the lower bound can be achieved in the case of the "largely" linear MINLP problem

$$
\begin{aligned}
\min Z = a^{\mathrm{T}}w + r(v) + c^{\mathrm{T}}y \\
\text{s.t. } Dw + t(v) + Cy \le 0 \qquad \text{(PL)} \\
Fw + Gv + Ey \le b \\
w \in W, \qquad v \in V, \qquad y \in Y
\end{aligned}
$$

where (w, v) are continuous variables and $r(v)$ and $t(v)$ are nonlinear convex functions. As shown by Quesada and Grossmann (1992), linear approximations to the nonlinear objective and constraints can be aggregated with the following MILP master problem:

$$\min Z_L^K = a^{\mathrm{T}}w + \beta + c^{\mathrm{T}} \qquad \text{(M-MIPL)}$$

$$\text{s.t. } \beta \geq r(v^k) + (\lambda^k)^{\mathrm{T}}\, [Dw + t(v^k) + Cy] - (\mu^k)^{\mathrm{T}}\, G(v-v^k) \qquad k = 1, \ldots, K$$

$$Fw + Gv + Ey \leq b$$

$$w \in W, \qquad v \in V, \qquad y \in Y, \qquad \beta \in R^l$$

Numerical results have shown that the quality of the bounds is not greatly degraded with the above MILP, as might happen if GBD is applied to (PL).

3. Incorporating Cuts

One way to expedite the convergence in the OA and GBD algorithms when the discrete variables in problem (P1) are 0–1 is to introduce the following integer cut, which has as an objective to make infeasible the choice of the previous 0–1 values generated at the K previous iterations (Duran and Grossmann, 1986c):

$$\sum_{i \in B^k} y_i - \sum_{i \in N^k} y_i \leq |B^k| - 1 \qquad k = 1, \ldots, K \qquad \text{(INTCUT)}$$

where $B^k = \{i \mid y_i^k = 1\}$, $N^k = \{i \mid y_i^k = 0\}$, $k = 1, \ldots, K$. This cut becomes very weak as the dimensionality of the 0–1 variables increases. However, it has the useful feature of ensuring that new 0–1 values are generated at each major iteration. In this way, the algorithm will not return to a previous integer point when convergence is achieved. With the above integer cut, the termination takes place as soon as ${Z_L}^K \geq \mathrm{UB}^K$. Also, in the case of the GBD method it is sometimes possible to generate multiple cuts from the solution of an NLP subproblem in order to strengthen the lower bound (Magnanti and Wong, 1981).

4. Handling of Equalities

For the case when linear equalities of the form $h(x, y) = 0$ are added to (P1), there is no major difficulty since these are invariant to the linearization points. If the equations are nonlinear, however, there are two difficulties. First, it is not possible to enforce the linearized equalities at K points. Second, the nonlinear equations may generally introduce nonconvexities. Kocis and Grossmann (1987) proposed an equality relaxation strategy in which the nonlinear equalities are replaced by the inequalities

$$T^k \nabla h(x^k, y^k)^{\mathrm{T}} \begin{bmatrix} x-x^k \\ y-y^k \end{bmatrix} \leq 0 \qquad (31)$$

where $T^k = \{t_{ii}^k\}$, and $t_{ii}^k = \text{sign}\,(\lambda_i^k)$ in which λ_i^k is the multiplier associated with the equation $h_i(x, y) = 0$. Note that if these equations relax as the inequalities $h(x, y) \leq 0$ for all y, and $h(x, y)$ is convex, this is a rigorous procedure. Otherwise, nonvalid supports may be generated. Also, note that in the master problem of GBD, (M-GBD), no special provision is required to handle equations since these are simply included in the Lagrangian cuts of (M-GBD). However, difficulties similar to those in OA arise if the equations do not relax as convex inequalities.

5. *Handling of Nonconvexities*

When $f(x,y)$ and $g(x,y)$ are nonconvex, two difficulties arise. First, the NLP subproblems (NLP1), (NLP2), and (NLPF) may not have a unique local optimum solution. Second, the master problem (M-MIP) and its variants (e.g., M-MIPF, M-GBD, M-MIQP) do not guarantee a valid lower bound $Z_L{}^K$ or a valid bounding representation with which the global optimum may be cut off. Two approaches can be used to address this problem: either assume a special structure in the MINLP problem and rely on methods for global optimization (Floudas and Grossmann, 1994); otherwise, apply a heuristic strategy to try to reduce the effect of nonconvexities as much as possible. We will describe only the second approach here, with the objective of reducing the effect of convexities at the level of the MILP master problem.

Viswanathan and Grossmann (1990) proposed introducing slacks in the MILP master problem to reduce the likelihood of cutting off feasible solutions. This master problem (augmented penalty/equality relaxation) (APER) has the following form:

$$\min Z^K = \alpha + \sum_{k=1}^{K} [w_p^k p^k + w_q^k q^k] \qquad \text{(M-APER)}$$

$$\text{s.t.}\ \left.\begin{array}{l} \alpha \geq f(x^k,y^k) + \nabla f(x^k,y^k)^{\mathrm{T}}\begin{bmatrix} x-x^k \\ y-y^k \end{bmatrix} \\ T^k \nabla h(x^k,y^k)^{\mathrm{T}}\begin{bmatrix} x-x^k \\ y-y^k \end{bmatrix} \leq p^k \\ g(x^k,y^k) + \nabla g(x^k,y^k)^{\mathrm{T}}\begin{bmatrix} x-x^k \\ y-y^k \end{bmatrix} \leq q^k \end{array}\right\} \quad k = 1, \ldots, K$$

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

$$x \in X, \qquad y \in Y, \qquad \alpha \in R^1, \qquad p^k,\ q^k \geq 0$$

where w_p^k, w_q^k are weights that are chosen sufficiently large (e.g., 1000 times magnitude of Lagrange multiplier). Note that if the functions are convex, the MILP master problem (M-APER) predicts rigorous lower bounds to (P1) since all the slacks are set to zero. It should also be noted that the program DICOPT++ (Viswanathan and Grossmann, 1990), which is currently the only MINLP solver that is commercially available (as part of GAMS; Brooke *et al.*, 1988), is based on the above master problem. This code also uses the relaxed (NLP1) to generate the first linearization for the above master problem, so the user need not specify an initial integer value. Also, since bounding properties of (M-APER) cannot be guaranteed, the search for nonconvex problems is terminated when there is no further improvement in the feasible NLP subproblems. Clearly, this is a heuristic, but one that works reasonably well in many problems.

It should also be noted that another modification to reduce the undesirable effects of nonconvexities in the master problem is to apply global convexity tests, followed by a suitable validation of linearizations. One possibility is to apply the test to all linearizations with respect to the current solution vector (y^K, x^K) (Kravanja and Grossmann, 1994). The convexity conditions that have to be verified for the linearizations are as follows:

$$\left.\begin{aligned} f(x^k,y^k) + \nabla f(x^k,y^k)^{\mathrm{T}}\begin{bmatrix} x_K - x^k \\ y_K - y^k \end{bmatrix} - \alpha \leq \epsilon \\ T^k \nabla h(x^k,y^k)^{\mathrm{T}}\begin{bmatrix} x_K - x^k \\ y_K - y^k \end{bmatrix} \leq \epsilon \\ g(x^k,y^k) + \nabla g(x^k,y^k)^{\mathrm{T}}\begin{bmatrix} x_K - x^k \\ y_K - y^k \end{bmatrix} \leq \epsilon \end{aligned}\right\} \quad k = 1, \ldots, K-1 \qquad \text{(GCT)}$$

where ϵ is a vector of small tolerances (e.g., 10^{-10}). Note that the test is omitted for the last linearizations since these are always valid for the last solution point (y^K, x^K). Based on this test, a validation of the linearizations is performed so that the linearizations for which the above verification is not satisfied are simply dropped out from the master problem. This test relies on the assumption that the solutions of the NLP subproblems are approaching the global optimum, and that the successive validations are progressively defining valid feasibility constraints around the global optimum. Also note that if the right-hand-side coefficients of linearizations are modified to validate the linearization, the test corresponds to the one in the two-phase strategy by Kocis and Grossmann (1988). Since in the latter case the violated linearizations are just shifted to widen the feasible region around the past solutions and not dropped out as in the former case, the nonconvexities tend to affect the MILP more than in the former case.

C. Logic-Based Methods

One current trend is to represent linear and nonlinear discrete optimization problems by models consisting of algebraic constraints, logic disjunctions, and logic relations (Balas, 1985; Beaumont, 1991; Raman and Grossmann, 1993, 1994). In particular, the mixed-integer program (P1) can also be formulated as a generalized disjunctive program, as has been shown by Raman and Grossmann (1994):

$$\begin{aligned} \min Z &= \sum_i \sum_k c_{ik} + f(x) \\ \text{s.t. } & g(x) \leq 0 \qquad \text{(DP1)} \\ \bigvee_{i \in D_k} & \begin{bmatrix} Y_{ik} \\ h_{ik}(x) \leq 0 \\ c_{ik} = \gamma_{ik} \end{bmatrix} \quad k \in SD \\ & \Omega(Y) = \text{true} \\ x \in R^n, \quad & c \in R^m, \quad Y \in \{\text{true, false}\}^m \end{aligned}$$

in which Y_{ik} are the Boolean variables that establish whether a given term in a disjunction is true $[h_{ik}(x) \leq 0]$ or false $[h_{ik}(x) > 0]$, while $\Omega(Y)$ are logical relations assumed to be in the form of propositional logic involving only the Boolean variables. Y_{ik} are auxiliary variables that control the part of the feasible space in which the continuous variables x lie, and the variables c_{ik} represent fixed charges which are activated to a value γ_{ik} if the corresponding term of the disjunction is true. Finally, the logical conditions, $\Omega(Y)$, express relationships between the disjunctive sets. In the context of synthesis problems the disjunctions in (DP1) typically arise for each unit i in the following form:

$$\begin{bmatrix} Y_i \\ h_i(x) \leq 0 \\ c_i = \gamma_i \end{bmatrix} \bigvee \begin{bmatrix} \neg Y_i \\ B^i x = 0 \\ c_i = 0 \end{bmatrix} \qquad (32)$$

in which the inequalities h_i apply and a fixed cost γ_i is incurred if the unit is selected (Y_i); otherwise $(\neg Y_i)$, there is no fixed cost and a subset of the x variables is set to zero with the matrix B^i. An important advantage of the above modeling framework is that there is no need to introduce artificial parameters for the "big-M" constraints that are normally used to model disjunctions. In the latter part of the next section, we will discuss methods for solving problems that are posed in logic form.

When the nonlinear discrete optimization problem is formulated as the generalized disjunctive program in (DP1), one can develop a corresponding logic-based branch-and-bound method. The basic difference is that the branching is performed directly on the disjunctions. While in the linear case branch-and-bound algorithms for solving (DP1) can be attractive (e.g., see Beaumont, 1991; Raman and Grossmann, 1994), for the nonlinear case they present some difficulties: first, because it is generally not trivial to develop a valid surrogate for the disjunctions that will effectively bound the relaxed solution (problem (DP1) without disjunctions); and second, because it is often not possible to reduce the dimensionality of the relaxed NLP subproblems. This can cause difficulties in structural flowsheet optimization problems as "dry" units with zero flow must then be handled, which often leads to singularities. Therefore, we consider only the corresponding OA and GBD algorithms for problems expressed in the form of (DP1).

As described in Turkay and Grossmann (1994), for fixed values of the Boolean variables, $Y_{\hat{i}k}$ = true and Y_{ik} = false, the corresponding NLP subproblem is as follows:

$$
\begin{aligned}
&\min Z = \sum_i \sum_k c_{ik} + f(x) \\
&\text{s.t. } g(x) \le 0 \qquad\qquad\qquad\qquad \text{(NLPD)} \\
&\left\{\begin{matrix} c_{\hat{i}k} = \gamma_{\hat{i}} \\ h_{ik}(x) \le 0 \end{matrix}\right\} \quad \text{for } Y_{\hat{i}k} = \text{true} \qquad k \in SD \\
&c_{ik} = 0 \quad \text{for } Y_{ik} = \text{false} \qquad i \ne \hat{i} \\
&x \in R^n, \qquad c \in R^m
\end{aligned}
$$

Note that only constraints corresponding to Boolean variables that are true are imposed. Also, fixed charges γ_{ik} are only applied to these terms. Assuming that K subproblems (NLPD) are solved in which sets of linearizations $l = 1,\ldots,K$ are generated for subsets of disjunction terms $L(ik) = \{l \mid Y^l ik = \text{true}\}$, one can define the following disjunctive OA master problem:

$$
\begin{aligned}
&\min Z = \sum_i \sum_k c_{ik} + \alpha \\
&\left.\begin{aligned} \text{s.t. } \alpha &\ge f(x^l) + \nabla f(x^l)^{\mathrm{T}}(x - x^l) \\ g(x^l) &+ \nabla g(x^l)^{\mathrm{T}}(x - x^l) \le 0 \end{aligned}\right\} \quad l = 1, \ldots, K \qquad\qquad \text{(MDP1)} \\
&\bigvee_{i \in D_k} \left[\begin{matrix} Y_{ik} \\ h_{ik}(x^l) + \nabla h_{ik}(x^l)^{\mathrm{T}} (x - x^l) \le 0 \qquad l \in L(ik) \\ c_{ik} = \gamma_{ik} \end{matrix}\right] \quad k \in SD \\
&\Omega\,(Y) = \text{true} \\
&\alpha \in R, \quad x \in R^n, \quad c \in R^m, \quad Y \in \{\text{true, false}\}^m
\end{aligned}
$$

It should be noted that before applying the above master problem, it is necessary to solve various subproblems (NLPD) so as to produce at least one linear approximation of each of the terms in the disjunctions. As shown by Turkay and Grossmann (1994), selecting the smallest number of subproblems amounts to the solution of a set-covering problem. In the context of flowsheet synthesis problems, another way of generating the linearizations in (MDP1) is by starting with an initial flowsheet and suboptimizing the remaining subsystems as in the modeling/decomposition strategy (Kocis and Grossmann, 1989b; Kravanja and Grossmann, 1990).

The above problem (MDP1) can be solved by the methods described by Beaumont (1991) and by Raman and Grossmann (1994). It is also interesting to note that, for the case of flowsheet synthesis problems, Turkay and Grossmann (1994) have shown that if the convex hull representation of the disjunctions in (32) is used to convert (MPD1) into an MILP problem, then assuming $B^i = I$ and converting the logic relations $\Omega(Y)$ into the inequalities $Ay \leq a$, it becomes equivalent to the master problem of the modeling/decomposition strategy:

$$\min Z_{\mathrm{L}} = \alpha$$

s.t.

$$\left.\begin{cases} \alpha \geq \sum_i c_i + f(x^l) + \nabla f(x^l)^{\mathrm{T}}(x - x^l) \\ g(\mathrm{x}^l) + \nabla g(x^l)^{\mathrm{T}} (x - x^l) \leq 0 \end{cases}\right\} \quad l = 1, ..., L \qquad \text{(MIPDF)}$$

$$\nabla h_i(x^l)^{\mathrm{T}} x \leq [-h_i(x^l) + \nabla h_i(x^l)^{\mathrm{T}} x^l]\, y_i \qquad l \in K_L^i, \quad i \in D_k, \quad k \in \mathrm{SD}$$

$$Ay \leq a$$

$$x \in R^n, \qquad c \geq 0, \qquad y \in \{0, 1\}^m$$

in which the linearizations for the constraints $h_i(x) \leq 0$ are ‘‘deactivated’’ when $y_i = 0$. Also, Turkay and Grossmann (1994) have shown that, while a logic-based GBD method cannot be derived as in the case of the OA algorithm, one can nevertheless determine for the MILP version of the master problem (MIPDF) one Benders iteration, which then yields a sequence similar to the GBD method for the algebraic case. Finally, it should also be clear that slacks can be introduced to (MDP1) and to (MIPDF) to reduce the effect of nonconvexities as in the augmented-penalty MILP master problem (M-APER).

D. Computational Example

In order to provide some insight into the computational performance of the MINLP algorithms described in the last section, we consider the following example problem:

$$
\begin{aligned}
\min Z &= y_1 + 1.5y_2 + 0.5y_3 + x_1^2 + x_2^2 \\
\text{s.t.} \quad & (x_1 - 2)^2 - x_2 \le 0 \\
& x_1 - 2y_1 \ge 0 \\
& x_1 - x_2 - 4(1 - y_2) \le 0 \\
& x_1 - (1 - y_1) \ge 0 \\
& x_2 - y_2 \ge 0 \\
& x_1 + x_2 \ge 3y_3 \\
& y_1 + y_2 + y_3 \ge 1 \\
& 0 \le x_1 \le 4, \qquad 0 \le x_2 \le 4 \\
& y_1, y_2, y_3 = 0, 1
\end{aligned}
\tag{33}
$$

Note that the nonlinearities involved in problem (33) are convex. Figure 11 shows the convergence of the OA and the GBD methods to the optimal solution using as a starting point $y_1 = y_2 = y_3 = 1$. The optimal solution is $Z = 3.5$, with $y_1 = 0$, $y_2 = 1$, $y_3 = 0$, $x_1 = 1$, $x_2 = 1$. Note that the OA algorithm requires three major iterations, while GBD requires four, and that the lower bounds of OA are much stronger.

Table II presents a set of 15 test problems that were used in the comparison of GBD with OA as implemented in DICOPT (Kocis and Grossmann, 1989a)

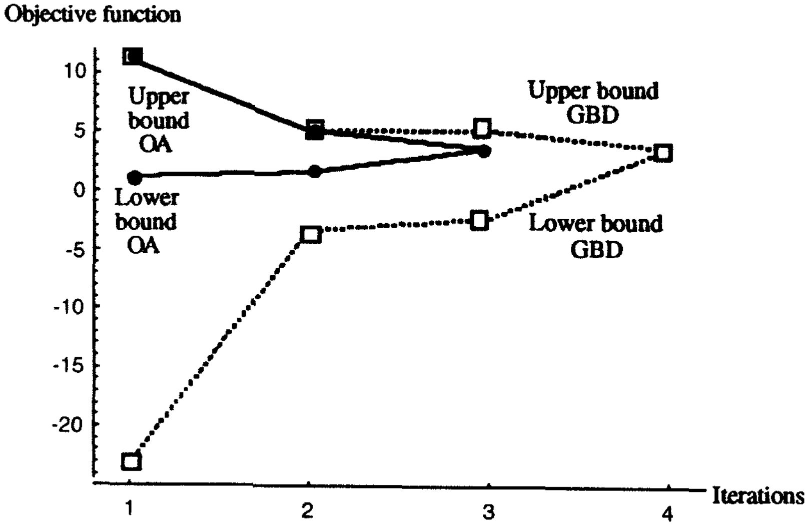


FIG. 11. Progress of iterations of OA and GBD for MINLP in problem (33).

TABLE II
TEST PROBLEMS FOR COMPARISON OF GBD, OA, AND AP/OA/RP

		Variables			Optimum value		
Problem	Type	Continuous	0–1	Total	Benders	DICOPT	DICOPT++
LAZAMY	Nonconvex	5	2	7	−333.89	−333.89	−333.89
HW3	Convex	2	3	5	3.50	3.50	3.50
HW74	Convex	7	3	10	−1.92	−1.92	−1.92
NONCON	Nonconvex	2	3	5	7.67	7.67	7.93
FLEX	Convex	11	4	15	−7.08	−7.08	−7.08
FRENCH	Convex	7	4	11	4.58	4.58	4.58
EX3	Convex	25	8	33	68.01	68.01	68.01
BATCHHW	Convex	10	9	19	106,755.80	106,756.00	106,755.80
TEST1A	Nonconvex	44	12	56	287,169.70	287,168.70	83,637.70
BATCH5	Convex	22	24	46	285,506.50	285,507.00	285,506.50
EX4	Convex	5	25	30	−8.06	−8.06	−8.06
REL1	Nonconvex	20	28	48	294.80	2,449.40	216.50
UTIL	Nonconvex	112	28	140	999.70	1,004.51	999.55
BATCH8	Convex	32	40	72	402,496.70	406,145.00	402,496.70
BATCH12	Convex	40	60	100	4,230,000	2,687,026	2,687,026

and with the augmented penalty/equality relaxation version of OA (AP/OA/RP) as implemented in DICOPT++ (Viswanathan and Grossmann, 1990), which is currently implemented in the modeling system GAMS. The size of these problems is relatively small. As seen in Table III, when there is a smaller number of 0–1 variables (less than 20), the performance of GBD is comparable to that of the OA methods. In fact, DICOPT++ requires more time because of the initialization step of solving the relaxed NLP subproblem. For a larger number of 0–1 variables (last five problems), the requirements of GBD greatly increase while the OA algorithms require a similar number of major iterations (between 3 and 5). Also, from Table II it can be seen that DICOPT (original OA) is most likely to fail to find the global optimum in nonconvex problems (e.g., REL1, UTIL), while DICOPT++ exhibits the greatest robustness in finding near-optimal solutions. It should also be noted that the CPU time in DICOPT is commonly the smallest because it often terminates prematurely owing to the higher lower bounds mistakenly predicted in nonconvex problems.

The solution of significantly larger MINLP problems has been reported in the literature. For instance, Viswanathan and Grossmann (1990) reported the solution with DICOPT++ of the superstructure optimization of the HDA process (see Section VIII, Fig. 17) which involved 13 0–1 variables, 709 continuous variables, and 719 constraints. DICOPT++ obtained the solution in three major iterations requiring 482 seconds on an IBM-3090. Also, Viswanathan and Gross-

TABLE III
Test Problems for Comparison of GBD, OA, and OP/OA/RP

	Iterations			CPU Time		
Problem	Benders	DICOPT	DICOPT++	Benders	DICOPT	DICOPT++
LAZAMY						
HW3	3	2	1	4.54	1.40	2.26
HW74	4	3	3	14.73	2.24	9.76
NONCON	4	2	4	3.48	1.58	4.32
FLEX	2	2	1	1.52	1.00	0.42
FRENCH	4	3	3	3.73	2.94	4.31
EX3	8	5	4	8.34	6.19	7.03
BATCHHW	8	3	5	10.94	4.00	45.56
TEST1A	5	3	3	5.62	3.68	34.60
BATCH5	2	2	4	14.97	11.88	93.96
EX4	67	4	3	766.88	33.62	26.94
REL1	70	3	5	394.81	41.51	100.87
UTIL	19	2	3	71.31	3.92	20.78
BATCH8	63	3	3	532.17	9.84	41.99
BATCH12	>66*	5	4	2527.20	141.61	108.58

*Search stopped after 66 iterations.

mann (1993) reported the solution with DICOPT++ of the optimal feed tray and number of plates in a distillation column for the separation of methanol and water. The model involved 115 0–1 variables, 1683 continuous variables, and 1919 constraints. DICOPT++ converged in seven major iterations and required about 70 minutes on an HP-9000/730 workstation. It is also interesting to note that significantly larger MILP problems have been solved in the area of scheduling (e.g., see Sahinidis and Grossmann, 1991a; Pinto and Grossmann, 1994).

VII. Solution Strategies for MINLP Synthesis Problems

In principle, the solution of an MINLP synthesis model can be obtained with the algorithms described in the previous section (e.g., OA or GBD algorithms). There are, however, three major difficulties that have to be addressed when attempting to solve these problems:

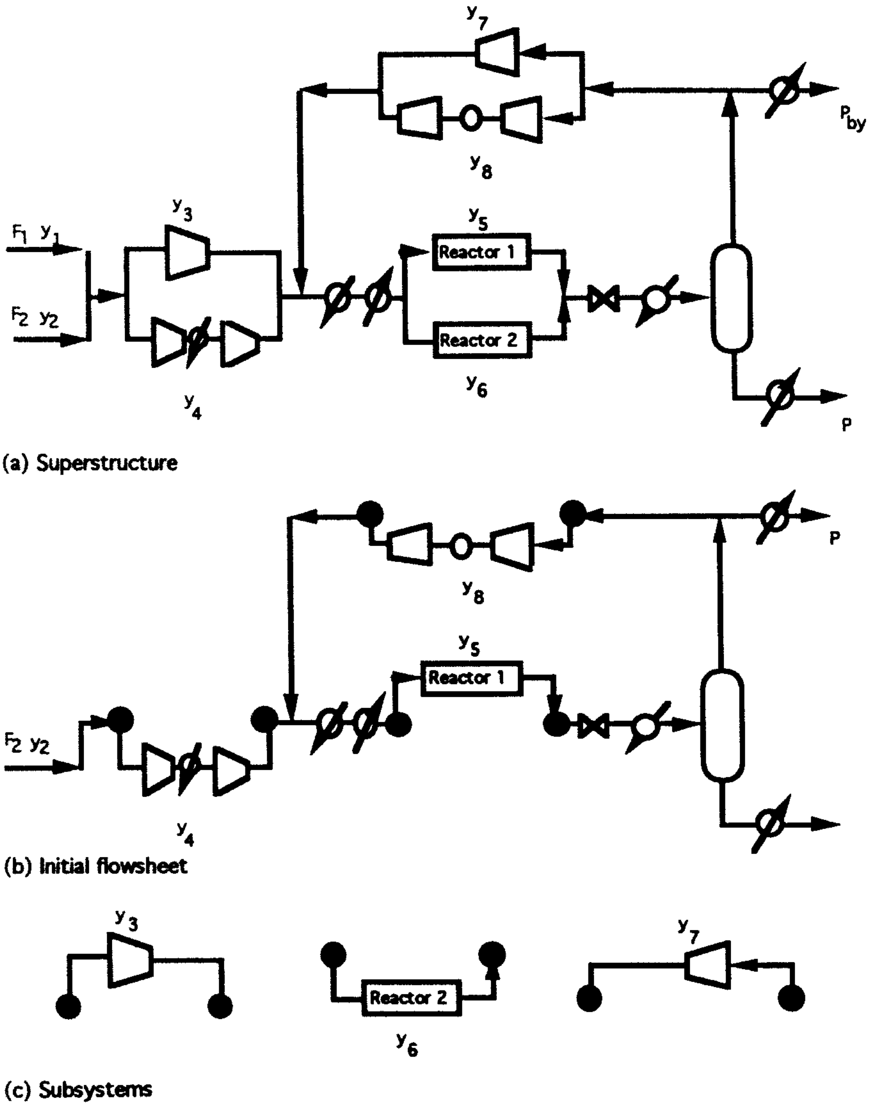
- Zero flows in process flowsheets
- Large size of MINLP problems
- Nonconvexities

A. Handling Zero Flows

The modeling/decomposition (M/D) strategy proposed by Kocis and Grossmann (1989b) is largely motivated by the need to simplify the solution of the NLP and MILP problems. It reduces the undesirable effect of nonconvexities and eliminates the optimization of "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, instead of optimizing it within the superstructure, as implied by problem (NLP2). The MILP solution is simplified by incorporating an approximation to the particular flowsheet only at each iteration. Finally, the effect of nonconvexities is reduced by special modeling techniques.

In the M/D strategy the basic idea 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 brief, the modeling is then performed as follows. Since interconnection nodes play a crucial role in defining the flowsheet structures and since they exhibit well defined equations, special modeling techniques can be applied to these nodes. In particular, splitters and mixers that imply the choice of a single alternative can be modeled through linear constraints, thus avoiding the nonconvexities associated with the use of split fractions (see (41)–(42) in

Section VII.C). 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. Finally, 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.



(a) Superstructure

(b) Initial flowsheet

(c) Subsystems

FIG. 12. Initial flowsheet and subsystems in modelling/decomposition strategy.

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 we are able to obtain linear approximations for the master problem of the existing process units. The question is, then, how to generate an approximation 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 provides not only nonzero flow conditions, but also points that are often good for approximating these units. An example of how a superstructure can be decomposed into subsystems to be suboptimized is given in Fig. 12 for the structural optimization of a process flowsheet. In this way, by optimizing the first flowsheet structure and suboptimizing the groups of nonexisting units, it suffices in subsequent iterations, to optimize the specific flowsheet that is generated in order to update the MILP. This has two desirable effects: it solves the NLP for each specific flowsheet, and it reduces the size of the MILP since only linearizations of existing units are incorporated at each iteration. This strategy has been automated in the flowsheet synthesizer PROSYN by Kravanja and Grossmann (1990), where the heat integration is handled through an extension of the constraints proposed by Duran and Grossmann (1986a), where area is accounted for with a variable HRAT. It is also important to note that if the MINLP were to be formulated as the generalized disjunctive programming problem (DP1), the logic-based algorithms would essentially reduce to the modeling/decomposition strategy, as has been discussed by Turkay and Grossmann (1994).

B. Large Size of MINLP Problems

There are several ways to avoid the solution of large MINLP synthesis problems. One is to rely on a targeting approach in which the solution of the MINLP is avoided by considering a sequence of design targets that give rise to aggregated models which are simpler to solve. As an example, consider the superstructure for heat exchanger networks in Fig. 4 (Section IV). Instead of solving directly the MINLP associated with this superstructure, consider the following stategy by Floudas *et al.* (1986), which has been implemented in the code MAGNETS by Ciric (1986) and which mimics the pinch design approach (Linnhoff, 1993):

(1) Determine the minimum utility solving the LP transshipment problem (see Fig. 7, Section IV).

(2) Given the energy target in step 1, determine the fewest number of units solving the MILP version of the transshipment model.

(3) Given the energy target of step 1 and the units predicted in step 2, determine the areas and flows of the superstructure in Fig. 4.

Conceptually, the disadvantage of this approach is that it is not entirely rigorous as the energy consumption and the number of units and areas are not optimized simultaneously. On the other hand, the advantage is that it involves the solution of simpler and smaller problems, which are computationally manageable.

Another approach that addresses the reduction of size of the MINLP is the state space approach by Bagajewicz and Manousiouthakis (1992). The basic idea of this strategy is to partition the synthesis problem into two major subsystems, the distribution network and the state space operator. The objective in the former is to make the decisions related to the distribution of flows in the superstructure, while the objective in the latter is to perform the optimization for the decisions selected in the distribution network. At the level of the state space operator one can consider the process either in its detailed level or simply as a pinch-based targeting model. While this strategy has the advantage of reducing the size of the MINLP, it is unclear how to develop automated procedures based on this approach.

A third option to avoid solving a large MINLP problem is to perform preliminary screening to eliminate some of the alternatives that are embedded in the superstructure, thus reducing the size of the MINLP problem. Daichendt and Grossmann (1994a,b) have proposed a design strategy for rigorously integrating preliminary screening and MINLP optimization. The strategy considers the synthesis problem as one in which the superstructure with its corresponding MINLP model is given. Rather than solving the entire MINLP model, the objective is to perform preliminary screening to rigorously eliminate a subset of suboptimal alternatives from the original superstructure to yield a *reduced superstructure*. The corresponding reduced MINLP model is to have the same global optimal solution as the original MINLP model.

The original superstructure is assumed to consist of a set of units $j \in U$, in which binary variables y_j are associated with the existence of these units and continuous variables d_j represent the corresponding sizes of these units. Additional continuous variables are represented by the vector x, corresponding to the state variables of the model, and by the vector z, corresponding to the continuous decision variables that incur linear costs in the objective function (e.g., raw material flows and energy consumption).

The MINLP problem (P1), corresponding to the original superstructure, is assumed to have the following form:

$$\min C = \sum_{j\in U} [a_j y_j + f_j(d_j)] + \sum_{k\in K} c_k z_k \qquad \text{(PP1)}$$

$$\text{s.t. } h_n(y,d,x,z) \le 0, \qquad n \in N$$

$$d_j \le d_j^{\mathrm{UB}}\, y_j, \qquad j \in U$$

$$y \in Y, \qquad d \in D, \qquad z \in Z, \qquad x \in X$$

where C is the cost function consisting of a nonlinear fixed charge function for the structural and design variables, y_j and d_j, and linear operating costs and revenues for the continuous variables z_k. The equations for mass and energy balances and inequalities for specifications are represented by the constraints $h_n(y,d,x,z) \leq 0$, $n \in N$. The inequality $d_j \leq d_j^{\mathrm{UB}}\, y_j$, $j \in U$, is a logical condition that will force the design variable d_j to be zero when y_j is zero, $\mathrm{d}_j^{\mathrm{UB}}$ being a valid upper bound to d_j. Finally, Y is a set of pure 0–1 constraints, $Y = \{y \mid Ay \leq a,\ y \in \{0,1\}^m\}$; and D, X, and Z are bounded sets in the space of positive real variables.

The strategy for preliminary screening relies on the use of an aggregated model that generally has a dimensionality less than or equal to that of the original model, and that may aggregate the units of the original model into subtasks. The aggregated model bounds the original model: The costs determined by the aggregated model underestimate the costs determined by the original model; the feasible region defined by the aggregated model overestimates the feasible region defined by the original model; and the aggregated model is formulated as a problem that can be solved to global optimality, i.e., as an MILP or convex MINLP problem (PP2), which is defined as follows:

The continuous variables of the aggregated model, $\hat{d}$, $\hat{x}$, and $\hat{z}$, are subvectors of the continuous variables d, x, and z of the original problem:

$$d = \begin{bmatrix} \hat{d} \\ \bar{d} \end{bmatrix}, \qquad x = \begin{bmatrix} \hat{x} \\ \bar{x} \end{bmatrix}, \qquad z = \begin{bmatrix} \hat{z} \\ \bar{z} \end{bmatrix}$$

where the variables with the bar are the ones excluded from the model. The binary variables of the aggregated model, $\hat{y}$, obey the following logic equivalence relationship (when treated as Boolean variables):

$$\hat{y}_i \Leftrightarrow \bigvee_{j \in U(i)} y_j, \qquad i \in \hat{U} \tag{34}$$

where $\Leftrightarrow$ and $\bigvee$ represent the logical equivalence and OR operator, and $U(i) \subseteq \hat{U}$ are subsets of units in the superstructure that are aggregated into subtasks $i \in \hat{U}$. As an example, consider the case of the HENS problem. Hot and cold stream *matches* are represented by the binary variables $\hat{y}_{h,c}$, $h \in H$, $c \in C$. These correspond to aggregated variables of the binary variables $y_{h,c,k}$ that represent the *units* for matches between streams h and c in stage k of the superstructure. It should be noted that when no unit aggregation takes place, $U = \hat{U}$, $|U(i)| = 1$, and $\hat{y}_i$ has a one-to-one correspondence with y_j.

The constraints of the aggregated model are linear combinations of subsets of the constraints of the original problem, $N(l) \subseteq N$:

$$r_l(\hat{y},\hat{d},\hat{x},\hat{z}\ ) = \sum_{n \in N(l)} \lambda_n^l h_n(y,d,x,z), \qquad l \in L \tag{35}$$

where L is the set of constraints of the aggregated model and λ_n^l are scalars for the linear combinations. To ensure bounding properties, the nonlinear fixed-charge cost functions of the original model will be *underestimated* by linear fixed-charge cost functions:

$$\sum_{i \subseteq \hat{U}} [b_i \hat{y}_i + g_i(\hat{d}_i)] \leq \sum_{j \subseteq U} [a_j y_j + f_j(\mathrm{d}_j)] \tag{36}$$

Because the aggregated model must rigorously bound the original model, it must be solvable to global optimality. In this paper the aggregated model is formulated as an MILP problem (PP2) having the following form:

$$\min \hat{\mathrm{C}} = \sum_{i \in \hat{U}} [b_i \hat{y} + g_i(\hat{d}_i)] + \sum_{k \in \hat{K}} c_k \hat{z}_k \tag{PP2}$$

$$\text{s.t. } r_l(\hat{y}, \hat{d}, \hat{x}, \hat{z}) \leq 0, \qquad l \in L$$

$$\hat{d}_i \leq \hat{d}_i^{\mathrm{UB}} \hat{y}_i, \qquad i \in \hat{U}$$

where $\hat{K} \subseteq K$ since $\hat{z}$ is a subvector of z. The sets $\hat{Y}$, $\hat{D}$, $\hat{X}$, $\hat{Z}$ are such that the feasible space $\hat{F}$ of (PP2) contains the feasible space of (PP1) projected in the space $(\hat{y}, \hat{d}, \hat{x}, \hat{z})$, i.e., $F^{\mathrm{p}} \subseteq \hat{F}$, where the projection of (PP1) is given by

$$F^{\mathrm{p}} = \{\hat{y}, \hat{d}, \hat{x}, \hat{z} \mid (y, d, x, z) \in F\} \tag{37}$$

Also, since the cost is underestimated, as given in (36), it follows that

$$\hat{C}^* \leq C^* \tag{38}$$

where C^* and $\hat{C}^*$ are the optimal solutions to (PP1) and (PP2), respectively. Thus, since (PP2) can be solved to global optimality because of its formulation as an MILP (or convex MINLP) problem, its solution provides a lower bound to (PP1).

It should be noted that deriving an aggregated model of the form of problem (PP2) is, in general, a nontrivial task, as it requires a basic understanding of the process as well as a characterization of the nature of the functions that are involved. Thus, problem (PP2) should be viewed as a conceptual framework rather than a recipe.

The solution to (PP2) yields a valid lower bound to the optimal solution of the original MINLP problem (PP1). The efficiency of the preliminary screening is clearly related to the tightness of the over- and underestimations; i.e., the closer the representation of the aggregated model is to the original model, the greater is the potential for problem size reduction. The purpose of the aggregated model is to generate successive solutions to (PP2) through integer cuts, whose costs are constrained to be less than or equal to the cost of a base-case design. The base-case design is a particular selection of units whose cost is determined using the original model. Since the base-case design is a feasible solution to (PP1), it provides a valid upper bound to the global optimal solution of (PP1) as well as a valid upper bound for the optimal solution for each iteration of (PP2), because (PP2) is always an underestimation of (PP1). Thus, when no additional feasible solution can be found to (PP2), preliminary screening is terminated.

Those units or subtasks that are not selected in any of the solutions are excluded from the superstructure. Those units that are selected in all solutions are permanently fixed in the superstructure; or, if the original units are aggregated into subtasks, constraints are imposed on the units that must be selected. Only those combinations of remaining units or subtasks that are feasible solutions to (PP2) are considered as alternatives for (PP1). The exclusion of certain units and the fixing of other units or combinations of units produces the reduced superstructure.

The base-case design can be determined either from the first solution to (PP2) or by selecting units based on heuristics. These are then fixed in (PP1), which is solved as an NLP problem. The closer the solution of the base-case design is to the global optimal solution of (PP1), the fewer solutions that result from (PP2). Thus, the efficiency, but not the rigorousness, of the preliminary screening procedure is dependent on the determination of a good base-case design. It should be noted also that work by the authors is currently under way to integrate preliminary screening and MINLP optimization within a hierarchical decomposition framework.

C. Handling Nonconvexities

One of the difficulties in the application of NLP and MINLP optimization techniques in process synthesis has been the fact that these problems are often nonconvex, and hence give rise to multiple local solutions. Although such techniques as simulated annealing might help to locate global or near-optimal solutions, they have the drawback that, aside from not being rigorous, they may require a very large number of trials, which in the long term will not help for handling process models that are relatively expensive to evaluate. Two major strategies for handling nonconvexities in synthesis problems are the reformulation of (MINLP) to an MILP model and the rigorous global optimization of nonlinear problems that exhibit special structure. We briefly describe these strategies here.

In order to derive an MILP approximation to problem (MINLP) (Papoulias and Grossmann, 1983c), the continuous variables x are partitioned as follows:

$$x = \begin{bmatrix} z^d \\ x^c \end{bmatrix} \tag{39}$$

in which z^d is the vector of operating conditions that gives rise to the nonlinearities (e.g. pressures, temperatures, split fractions, conversions, etc.), and x^c is a vector of material, heat and power flow variables. In this way, given a fixed value of z^d, the nonlinear equations, which are assumed to be only a function of x, reduce to a subset of linear equations; that is,

$$h(x) = 0 \quad \Rightarrow \quad Ex^c = \mathrm{e} \tag{40}$$

in which the matrix of coefficients E and the right-hand side e are functions of z^d, $E(z^d)$, $e(z^d)$.

Since, in general, more than one fixed value for the variables z^d are considered, this requires the introduction of the additional 0–1 variables y^d to represent the potential selection of the discrete operating conditions. In this way, the general form of the MILP approximation will be as follows:

$$\begin{gathered}\min C = a_1^{\mathrm{T}}y + a_2^{\mathrm{T}}y^d + b^{\mathrm{T}}x^c \\ \text{s.t. } E_1y^d + E_2x^c = e \\ D_1y + D_2y^d + D_3x^c \leq d \\ y,\ y^d = 0,1 \qquad x^c \geq 0\end{gathered} \tag{MAPP}$$

It should be noted that the derivation of the above problem generally requires the disaggregation of the vector of continuous variables x^c in terms of the discretized conditions. To illustrate this point more clearly, consider the simple splitter shown in Fig. 13. The corresponding mass balance equations for each component i are as follows:

$$f_i^1 = \eta\, f_i^{\mathrm{in}} \tag{41}$$

$$f_i^2 = f_i^{\mathrm{in}} - f_i^1 \tag{42}$$

where η is the split fraction for outlet stream 1. Note that Eq. (41) is nonlinear (in fact bilinear), and despite its simplicity it is a major source of nonconvexities and numerical difficulties.

Now assume that we consider N discrete values of η, η_k, $k = 1, 2, \ldots, N$. Then if we disagggregate the flow for the inlet stream as $f_i^{\mathrm{in},k}$, $k = 1, 2, \ldots, N$, and introduce the 0–1 variables $y^{d,k}$, $k = 1, 2, \ldots, N$, equations (41) and (42) can be replaced by the linear constraints

$$f_i^1 = \sum_{k=1}^{N} \eta_k f_i^{\mathrm{in},k} \tag{43}$$

$$f_i^{\mathrm{in},k} - Uy^{d,k} \leq 0 \qquad k = 1, 2, \ldots, N \tag{44}$$

$$\sum_{k=1}^{N} y^{d,k} = 1 \tag{45}$$

$$f_i^2 = f_i^{\mathrm{in}} - f_i^1 \tag{46}$$

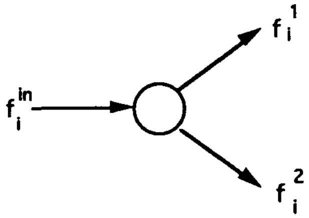


FIG. 13. Stream splitter.

While the nonlinearities are eliminated, it is clear the number of discrete and continuous variables is increased as well as the number of constraints. Also, in the general case the definition of the matrix of coefficients and the right-hand sides of problem (MAPP) requires an *a priori* evaluation or simulation of nonlinear models.

Extensive reviews on global optimization can be found in Horst (1990) and Horst and Tuy (1990). In this section we present a summary of a global optimization method that has been developed by Quesada and Grossmann for solving nonconvex NLP problems which have the special structure that they involve linear fractional and bilinear terms. It should be noted that global optimization has clearly become one of the new trends in optimization and synthesis, and active workers involved in this area include Floudas and Visweswaran (1990), Swaney (1990), Manousiouthakis and Sourlas (1992), and Sahinidis (1993).

The special class of NLP problems considered by Quesada and Grossmann (1995b) has been motivated by heat exchange networks and separations problems. These can be represented in general as follows:

$$\begin{aligned} &\min g_0 \\ &\text{s.t. } g_l \le 0 \qquad l = 1, \ldots, L \end{aligned} \qquad \text{(NLP)}$$

where

$$g_l = \sum_{i \in I} \sum_{j \in J} c_{ijl} \frac{x_i}{y_j} - \sum_{i \in I'} \sum_{j \in J'} c_{ijl}\, x_i y_j + h_l(x, y, z), \qquad l = 0, 1, \ldots, L$$

$$x^{\mathrm{L}} \le x \le x^{\mathrm{u}}$$

$$y^{\mathrm{L}} \le y \le y^{\mathrm{u}}$$

$$z \in Z$$

As shown above, the objective function and the constraints generally involve linear fractional and bilinear terms corresponding to the two summation terms, while the last term $h_l(x, y, z)$ is assumed to correspond to a convex function. This type of problem arises, for instance, in the optimization of heat-exchanger networks (linear area cost, arithmetic mean driving force, isothermal mixing; see Quesada and Grossmann, 1993) and in the case of synthesis of sharp separation networks with mixed products (see Quesada and Grossmann, 1995a). The difficulty involved in solving these NLP optimization problems is that a straightforward application of common local search methods is generally not rigorous. Conventional NLP algorithms can not only produce local solutions that are suboptimal, but also may fail to find a feasible solution owing to the nonconvexities of the constraints.

In the method proposed by Quesada and Grossmann (1995b), the main idea is to replace the bilinearities and linear fractional terms by valid under- and over-estimators which will yield a convex NLP (or LP) whose solution provides a lower bound to the global optimum. Consider, for instance, fractional terms with positive coefficients. By introducing the variables r_{ij}, we can express the fractional term as the constraint

$$r_{ij}\, y_j \geq x_i \qquad i \in I,\, j \in J \tag{47}$$

which is nonconvex. Valid linear over-estimators, which were suggested by McCormick (1976) for this constraint, are given by,

$$x_i \leq y_j^{\mathrm{u}} r_{ij} + r_{ij}^{\mathrm{L}} y_j - y_j^{\mathrm{u}} r_{ij}^{\mathrm{L}} \qquad i \in I,\, j \in J \tag{48}$$

$$x_i \leq y_j^{\mathrm{L}} r_{ij} + r_{ij}^{\mathrm{u}} y_j - y_j^{\mathrm{L}} r_{ij}^{\mathrm{u}} \qquad i \in I,\, j \in J \tag{49}$$

where x_i^{L}, x_i^{u}, y_j^{L}, y_j^{u}, r_{ij}^{L}, r_{ij}^{u} are valid lower and upper bounds of the variables. In addition, Quesada and Grossmann (1995b) showed that the nonlinear convex constraint

$$r_{ij} \geq \frac{x_i}{y_j^{\mathrm{u}}} + x_i^{\mathrm{L}} \left(\frac{1}{y_j} - \frac{1}{y_j^{\mathrm{u}}} \right) \qquad i \in I,\, j \in J \tag{50}$$

can be used as a valid underestimator. The interesting feature of (50) is that it is a stronger constraint than (48) and (49), provided r_{ij}^{L}, r_{ij}^{u} are given by the bounds of x_i and y_j. In fact, when these bounds are obtained by the optimization of individual variables in (NLP), it is also possible to generate projected bounding constraints which can serve to tighten the representation of the NLP.

The proposed method then consists in reformulating problem (NLP) in terms of valid linear and nonlinear bounding constraints such as in (48)–(50), giving rise to a convex NLP (or LP) problem which predicts valid lower bounds to the global optimum. If there is a difference between the current upper and lower bounds, the idea is to partition the feasible region by performing a spatial branch-and-bound search. The major steps in the global optimization algorithm by Quesada and Grossmann (1995b) for NLP problems involving linear fractional and bilinear terms are as follows:

Step 0. Initialization step

(a) Set the upper bound to $f^* = \infty$; the tolerance ϵ is selected.

(b) Bounds over the variables involved in the nonconvex terms are obtained. For this purpose, specific subproblems can be solved or a relaxation of the original problem is used. Update the upper bound f^*.

(c) Define space W_0 as a valid relaxation of the feasible region in the space of the nonconvex variables. The branch-and-bound search will be conducted over W_0. The list F is initially defined as the region W_0.

(d) Construct a convex underestimator problem ($\mathrm{CU_L}$) by replacing the nonconvex terms in the original problem with additional variables and introducing valid convex approximations of these nonconvex terms. Valid constraints that were not present in the original problem because they were redundant can be included to tighten the convex relaxation.

Step 1. Convex underestimator problem

(a) Solve problem $\mathrm{CU_L}$ over the relaxed feasible region W_0. The solution corresponds to a valid lower bound (f^{L}) of the global optimum. The actual objective function is evaluated if this is a feasible solution; otherwise, the original problem is solved using the convex solution as the initial point. Update the upper bound.

(b) If $(f^* - f^{\mathrm{L}}) \leq \epsilon f^*$, stop; the global solution corresponds to f^*.

Step 2. Partition

From the list F, consider a subregion W_j (generally the region with the smallest f^{L} is selected) and divide it into two new subregions W_{j+1} and W_{j+2}; these are added to the list F and subregion W_j is deleted from F.

Step 3. Bounding

(a) Solve problem $\mathrm{CU_L}$ for the two new subregions.

(b) If the solutions are feasible, evaluate the actual objective function. Otherwise, the original nonconvex problem can be solved according to a given criterion.

Step 4. Convergence

Delete from list F any subregion with $(f^* - f^{\mathrm{L}}) \leq \epsilon f^*$. If list F is empty, stop; the global optimum is f^*. Otherwise, go to step 2.

The global optimization algorithm described above uses a spatial branch-and-bound procedure (steps 2 to 4). Like many branch-and-bound methods, the algorithm consists of a set of branching rules, together with upper bounding and lower bounding procedures.

The branching rules include the node selection rule, the branching variable selection, and the level at which the variable is branched. A simple branching strategy is as follows. The node with the smallest lower bound is the node selected to branch on, and two new nodes are generated using constraints of the type

$$x_i \geq x_i^* \qquad \text{and} \qquad x_i \leq x_i^* \tag{51}$$

Different strategies can be used to do the branching. These include generating more than two nodes from a parent node, using different types of branching constraints or different node selection rules. For the latter, we can use some type of degradation function similar to the one used in branch-and-bound procedures for MILP problems.

Additional criteria used in branch-and-bound algorithms for MILP problems can be extrapolated to the global optimization case. These include the fixing of variables, tightening of bounds, range reduction, etc. (see Sahinidis, 1993). One main difference between the branch-and-bound procedure used for binary variables and the spatial branch-and-bound search used here is the fact that it might be necessary to branch more than once on the same variable. When the selection rule involves more than one variable within a small range, it is often useful to branch on a variable that has not been used previously, even though it may not be the first candidate.

Information of the convex underestimator problem can be employed to select the branching variables. At this point, only the difference between the convex solution and the actual value of the functions is used. It is also possible to consider dual or second-order information or to generate small selection subproblems (Swaney, 1990). With respect to the upper bound, there are two cases. In the first case, when the feasible region of the original problem is convex, the evaluation of the original objective function at the solution of the convex underestimator problem often provides a good upper bound. In the second case, when the feasible region is nonconvex, it is sometimes necessary to obtain an upper bound through a different procedure, since the solution of the convex underestimator problems might be infeasible for the original problem. In some particular cases, it may be better to use a specialized heuristic to obtain a good upper bound. In general, however, it is necessary to solve the original nonconvex problem to generate an upper bound. As pointed out in Quesada and Grossmann (1995a,b) the solution of the convex underestimator problem provides a good initial point to the nonconvex problem.

VIII. Applications

As was described in the review of previous work, over the last ten years MINLP optimization models have been reported for the synthesis of process flowsheets, heat-exchanger networks, separation sequences, reactor networks, utility plants, and design of batch processes. Rather than describing in detail each of these works, we will briefly highlight several examples from our research group at Carnegie Mellon to illustrate the capabilities and the current limitations of the MINLP approach.

First, we present a small example from the MINLP model by Yee *et al.* (1990b) for the synthesis of a heat-exchanger network for the stream data given in Table IV. If the MINLP model (21)–(29) is solved with two stages (see Fig. 5) and with a code such as DICOPT++ (Viswanathan and Grossmann, 1990), we obtain the design given in Fig. 14. Note that that design requires neither heating nor cooling. On the other hand, the network involves stream splitting which is not always attractive from a practical point of view as this requires the additional investment of a control valve and a potentially more complex operation. One can easily generate a network structure with no stream splitting by adding the inequalities that allow at most one heat exchange for each stream in each stage. The resulting solution is shown in Fig. 15. Note that the new structure requires heating and cooling, although in small amounts. Nevertheless, the network now consists of four instead of three units, and the cost penalty for not having stream splits is of the order of 15% ($94,268/year vs. $82,491/year). This example shows that by imposing discrete constraints, one can synthesize several network structures with which one can assess different designs with other criteria such as complexity, operability, etc.

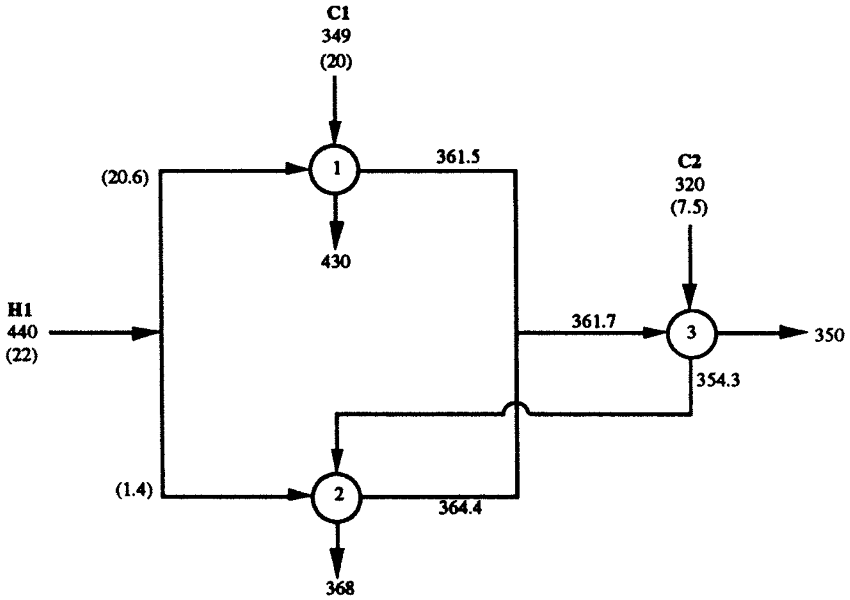
As a second example we present the synthesis of a 10-stream problem (5 hot and 5 cold), using the program SYNHEAT (Bolio *et al.*, 1993) which implements the MINLP method by Yee *et al.* (1990b) with DICOPT++. The size of the MINLP model is 135 0–1 variables, 376 continuous variables, and 536 constraints. The solution, which is shown in Fig. 16, required about 5 minutes on a VAX-6325. It should be noted that the solution to this problem was simplified by the various exchangers that are assigned only to utilities.

As a third example, consider the HDA process studied extensively by Douglas (1988). The superstructure for this process is shown in Fig. 17, which is based on a preliminary qualitative analysis of alternatives described in Douglas (1988). Given the basic options considered for the selection of reactors and the use of membrane separators, as well as a restricted set of alternatives for the

TABLE IV
DATA FOR ONE-HOT/TWO-COLD STREAM PROBLEM

Stream	TIN (K)	TOUT (K)	F_{cp} (kW/K)	h (kW/h m K)	Cost ($/kW-year)
H1	440	350	22	2.0	—
C1	349	430	20	2.0	—
C2	320	368	7.5	0.67	—
S1	500	500	—	1.0	120
W1	300	320	—	1.0	20

Note. Minimum approach of temperatures (EMAT) = 1K. Exchanger cost = 8,600 + 670 (Area)^0.83.



Total Heat Exchangers Area = 182.9 m2

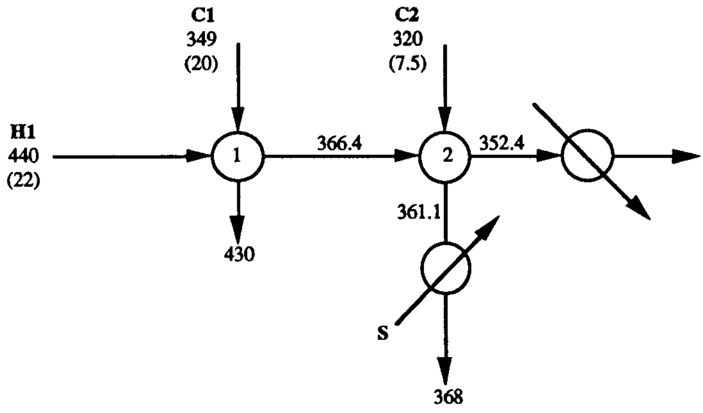
Utilities:
Heaters heat load = 0 Kw
Coolers heat load = 0 Kw

Costs:
Investment = $ 82, 491.6 per year

Total = $ 82, 491.6 per year

FIG. 14. Optimal network with no constraints on split streams.

separation and recycle, it is a relatively simple task to develop the superstructure representation, which has embedded close to 200 different flowsheet configurations. 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). Using the starting configuration shown in Fig. 18, the optimal solution, which is shown in Fig. 19, was obtained with both the M/D strategy and with the AP/OA/ER algorithm. Note that the profit is increased from $4,814,000/year to $5,887,000/year. The M/D strategy required 2 minutes of CPU time (IBM-3083), while AP/OA/ER required 8 minutes; both took two major iterations. This example shows that computational savings can be achieved with the M/D strategy and that the economic improvements obtained in structural optimization can be very large.



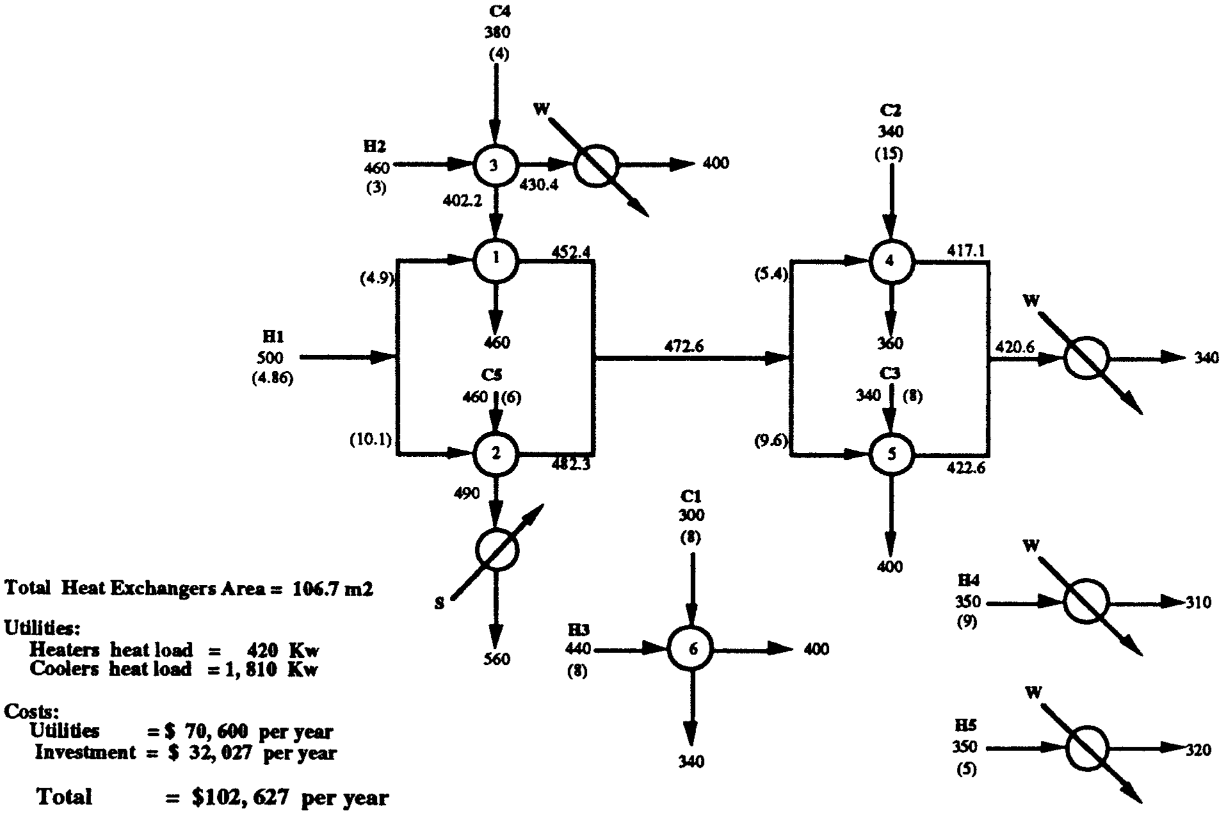
Total Heat Exchangers Area = 165.4 m2

Utilities:
Heaters heat load = 52.1 Kw
Coolers heat load = 52.1 Kw
Costs:

Utilities = $ 7,293.4 per year
Investment = $ 86,975.8 per year

FIG. 15. Network structure with no stream splits.

As a fourth example, consider the simultaneous optimization and heat integration of the flowsheet superstructure in Fig. 12 which has 16 embedded flowsheet alternatives. The M/D strategy implemented in PROSYN was applied to this problem. The resulting MINLP formulation contains 293 constraints, 279 continuous variables, and 8 binary variables. This example problem was solved with PROSYN for the three following cases: (a) MINLP optimization with no heat integration, (b) simultaneous MINLP optimization and heat integration using the model by Duran and Grossmann, (c) simultaneous NLP optimization and heat integration with HEN costs for the optimal structure obtained in case (b). For cases (a) and (b), the OA/ER algorithm was terminated based on the progress of the NLP solutions, since higher bounds on the profit were obtained from the MILP master with the proposed deactivation scheme for the linearizations of the splitter in the recycle. The OA/ER algorithm requires two NLP subproblems to confirm that the initial flowsheet in case (a) is the optimum. In case (b) it requires three NLP subproblems to find the structure in Fig. 19. This clearly indicates that the quality of the information supplied to the MILP master problem by the M/D strategy is good.



Total Heat Exchangers Area = 106.7 m2

Utilities:
Heaters heat load = 420 Kw
Coolers heat load = 1, 810 Kw

Costs:
Utilities = $ 70, 600 per year
Investment = $ 32, 027 per year

Total = $102, 627 per year

FIG. 16. Synthesis of a 10-stream problem with SYNHEAT.

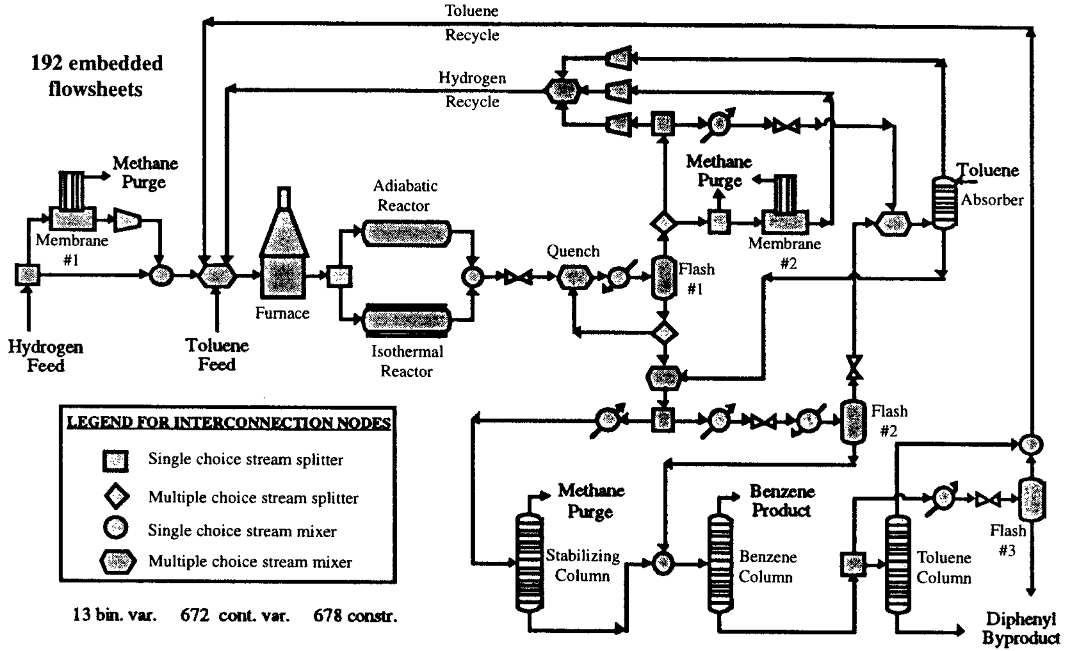


FIG. 17. Superstructure of HDA process.

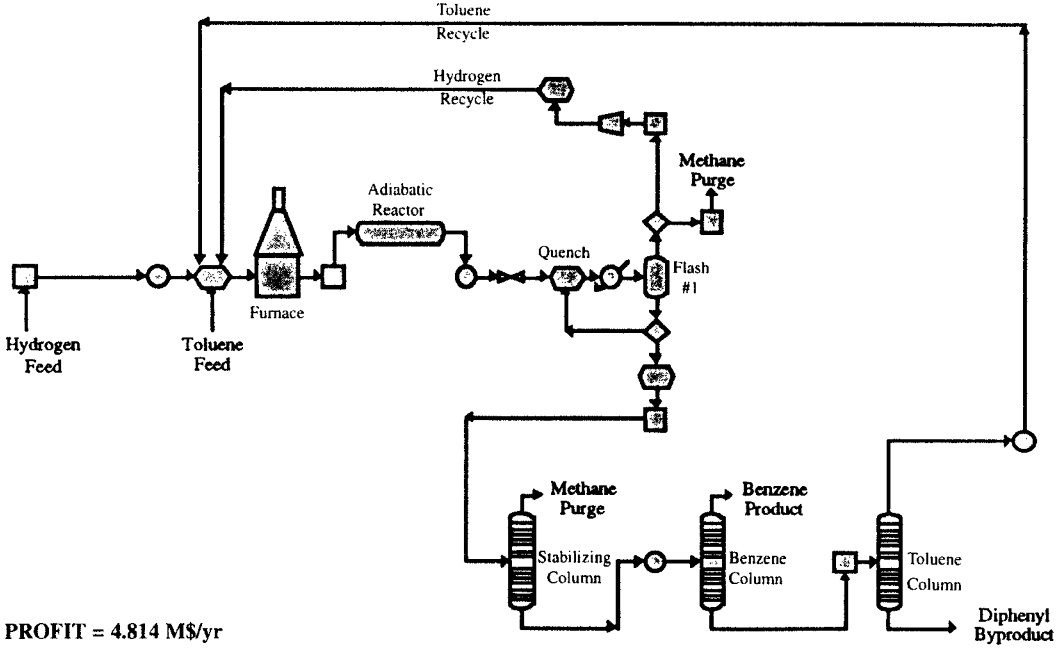


FIG. 18. Initial flowsheet for HDA process.

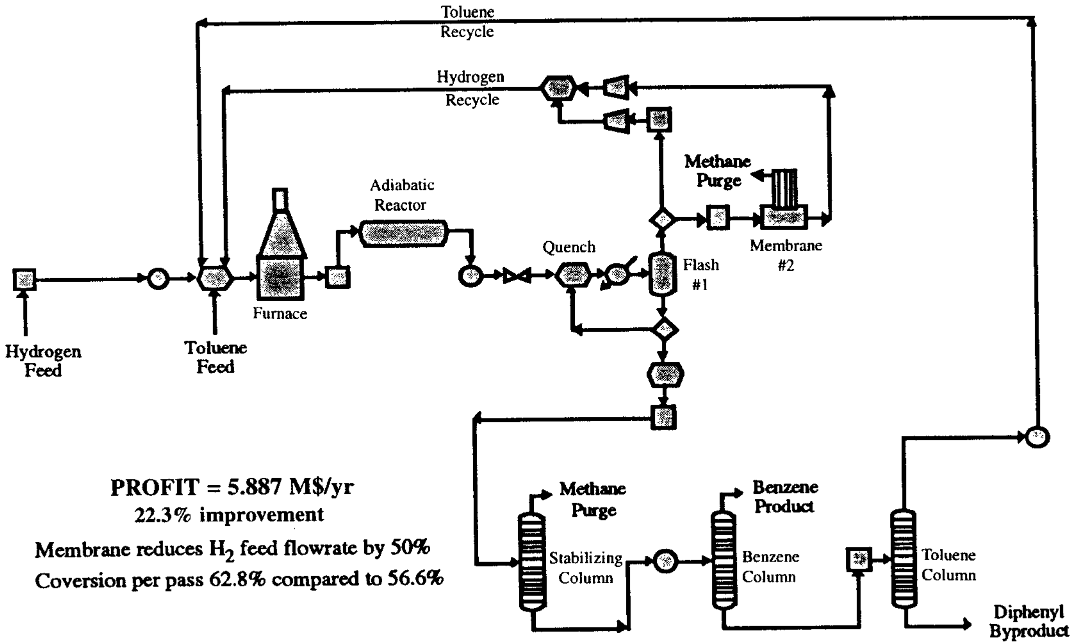


FIG. 19. Optimal flowsheet from HDA superstructure.

First consider case (a) when only the MINLP optimization of the superstructure is performed without heat integration. The optimal flowsheet is $y^k = \{1,0,0,1,1,0,0,1\}$ with annual profit of 794,000 \$/year. It utilizes the cheaper feedstock F1, two-stage feed compression, cheap reactor R1 with low conversion, and a two-stage compressor for the recycle. If costs of the HEN (which are quite significant) are subsequently calculated and added to the profit, the result is a loss of 1,192,000 \$/year. When heat integration is simultaneously performed in the MINLP optimization of the superstructure (Duran and Grossmann, 1986b), the results are much better at first glance. The optimal flowsheet in Fig. 20 yields an annual profit of 3,403,000 \$/year (2,609,000 \$/year more than for the nonintegrated flowsheet). The difference in the new flowsheet lies in the selection of single-stage compressors for the feed and the recycle. Also, almost all operating conditions change significantly since the trade-offs between heat integration (consumption of steam and cooling water), electricity, and consumption of feedstock are now appropriately established. As seen in Table V, energy is recovered within the process, so no expensive heating utility is required. The overall conversion of B is increased from 58.3 to 63.7%, and the reactor operates at 2.5 MPa instead of 7.05 MPa as in case (a). Because of the relatively small vertical driving forces and the gas–gas matches, however, the HEN costs are very high, so that annual profit when these costs are added to the expenses reduces −292,000 \$/year. In order to consider the HEN costs, the approach suggested by Kravanja and Grossmann (1990) was applied, yielding a profit of 1,679,000 \$/year. As can be seen from Table V, the operating conditions again undergo considerable changes. The most significant differences are a further increase in the overall conversion to 66.04%, elimination of the preheat of the reactor feed (gas–gas matches with small temperature driving forces), and

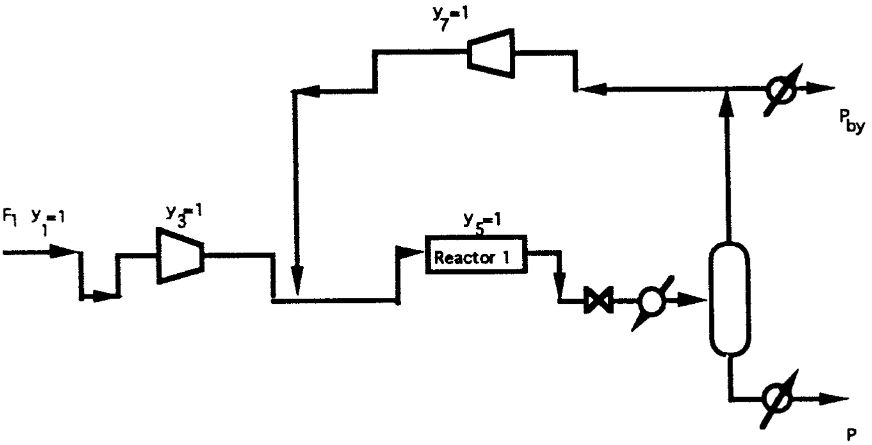


FIG. 20. Optimal flowsheet with heat integration from superstructure in Fig. 12.

TABLE V
TECHNICAL AND ECONOMIC RESULTS FOR EXAMPLE OF FIG. 20

	MINLP only	Heat integration Duran–Grossman	Heat integration HEN costs
Flows, kg-mol/s			
F1	6.176	5.648	5.451
F2	0	0	0
Purge rate, %	14.5	14.6	19.7
Reactor			
P_{in}, MPa	7.048	2.500	4.377
P_{out}, MPa	6.343	2.250	3.939
T_{out}, K	378	430	419
T_{in}, K	332	379	356
Conversion of B Per pass, %	25.5	25.4	29.4
Volume, m^3	55.7	49.1	67.7
Flash separation			
P, MPa	6.343	2.250	4.377
T_{out}, K	378	310	310
Utilities			
Electricity, MW	3.718	1.798	2.78
Heating, steam, 10^9 MJ/year	0.114	0	0
Cooling, water, 10^9 MJ/year	1.566	0.834	1.05
Other			
Overall conversion of B, %	58.92	63.7	66.04
Load of HEN, MW	54.9	71.5	48.0
Earnings, 10^3 \$/year			
Product	8000	8000	8000
Byproduct	1513	1341	1309
Expenses, 10^3 \$/year			
Feedstock capital investment	4632	4236	4088
HEN	1986	3695	1173
Other	1131	659	925
Electricity compress	948	459	709
Heating utility	912	0	0
Cooling utility	1096	584	735
Annual profit, 10^3 \$/year			
Without HEN costs	794	3403	2852
With HEN Costs	**−1192**	**−292**	**1679**

selection of the reactor pressure at 4.377 MPa, which lies between the pressures of cases (a) and (b). Note that the HEN costs are significantly reduced (see Fig. 21) while other capital and utility costs increased (electricity and cooling) to yield a profit increase of 2,871,000 \$/year when compared to case (a) where no heat integration was considered, and an increase of 1,971,000 \$/year compared to case (b). This clearly shows the importance of anticipating the heat integration in the synthesis stage with both utility and investment costs.

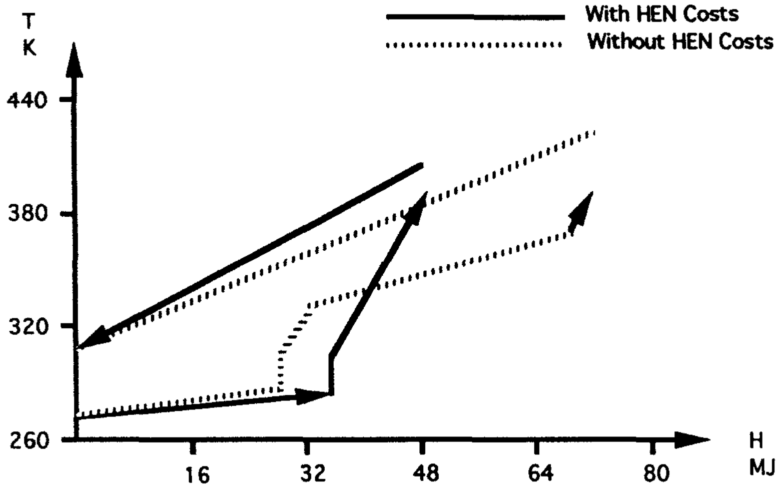


FIG. 21. Composite heating and cooling curves for alternative solutions.

As the next example, we consider the synthesis of a network of sharp separators for the specifications given in Fig. 22. The objective function considered is simply a linear function of the total flows. As shown in Quesada and Grossmann (1995a), the optimization of the corresponding superstructure corresponds to a nonconvex NLP with bilinear constraints (flows times compositions and flows times splits)—one for which a special LP underestimator function can be derived. This method, which has been automated in the program GLOBESEP developed by Bolio *et al.* (1994), produced the global optimum configuration given in Fig. 23. The global optimum solution with an objective value of 159.48 was determined by analyzing a total of seven nodes in the branch and bound search, solving seven LP underestimators and six NLPs. The global optimum was confirmed within a tolerance of less than 0.1%. It is interesting that the

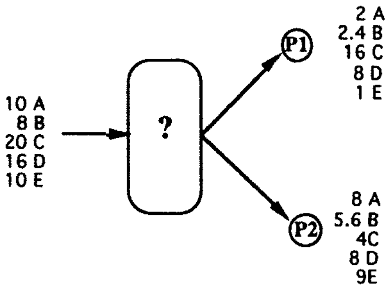


FIG. 22. Synthesis of 5-component separation network.

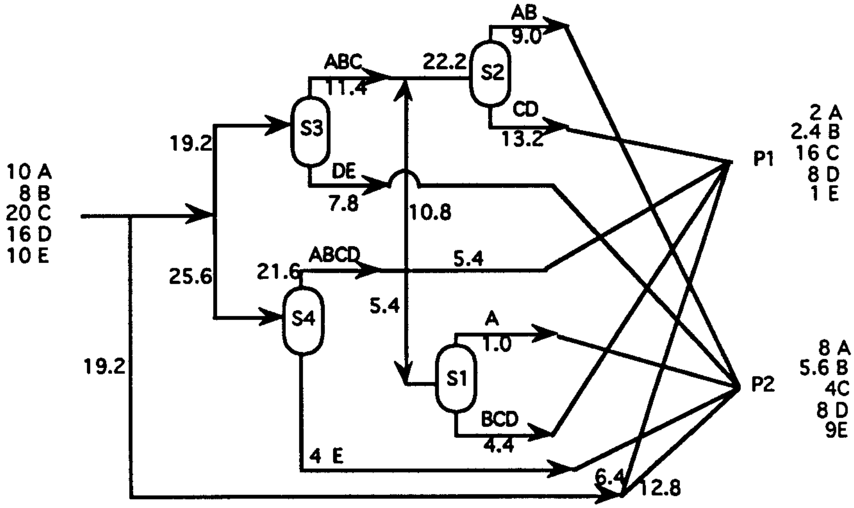


FIG. 23. Global optimum separation network with objective 159.48.

initial lower bound that is predicted is rather tight, and that there is at least a second local solution with objective 162.36. This problem, which involved 225 continuous variables, was solved with GLOBESEP in less than 10 seconds of CPU time on an IBM RS6000 using the GAMS/MINOS optimization software.

As the final example, we consider the preliminary screening procedure by Daichendt and Grossmann (1994a,b) that was applied to the MINLP model and superstructure for heat-integrated distillation sequences of Floudas and Paules (1988). The superstructure is given in Fig. 24. The problem involves the separation of a simple ternary mixture into 98%-pure components using sharp splits and adiabatic columns. The problem was aggregated into an MILP as described in the previous section. The base-case design is determined by the initial solution to the MILP, yielding a lower bound to the global optimal solution of the MINLP. Furthermore, the actual costs corresponding to the selection of units and operating conditions can readily be calculated, yielding an upper bound to the global optimal solution. The DICOPT++ implementation for solving directly the MINLP involves 18 binary variables, 77 continuous variables, and 143 constraints. This can be compared to the aggregated MILP model, which involves 18 binary variables, 117 continuous variables, and 225 constraints. The base-case solution consists of columns 1 and 3, with the distillate of column 3 (376 K) heat-integrated with the bottoms of column 1 (361 K). Low-pressure steam (421 K) is used to heat the reboiler of column 3 (396 K), while the less expensive exhaust steam (373 K) is used to supply the additional heat required

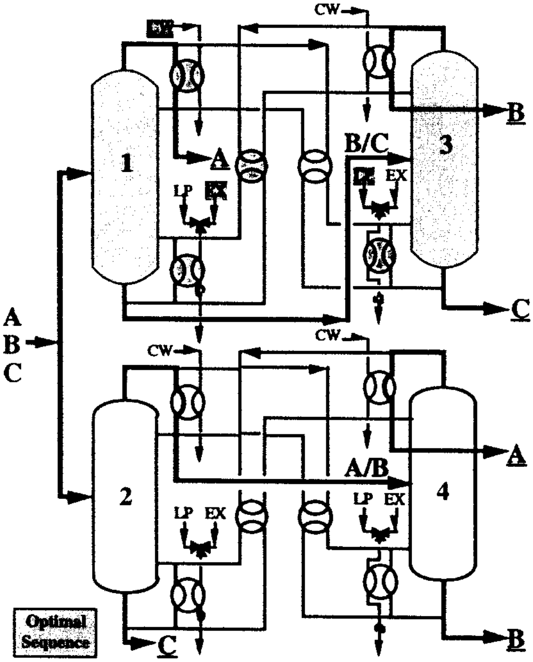


FIG. 24. Superstructure for heat-integrated distillation sequences.

for the reboiler of column 1. Cooling water (305–325 K) is used for the condenser of column 1 (342 K). The optimal sequence is shaded in Fig. 24. The lower bound to the global optimal solution is $464,500/year. The upper bound is determined to be $466,800/yr. Therefore, the global optimal solution is bounded to within 0.5%.

Solving this sequence as an NLP yields an optimal solution of $466,600/year. No feasible alternative solution to this problem could be found during preliminary screening, so the base-case design corresponds to the optimal configuration. In contrast, the full MINLP when solved with DICOPT++ converged to a suboptimal solution of $763,600/year (64% more expensive) using default NLP parameter settings, and $502,100/year (8% more expensive) using the modified NLP parameter settings. Although the DICOPT++ implementation does not converge to the same solution, we note that determining a base-case design and performing preliminary screening yielded a reduction of 41% compared to the computation time required with DICOPT++ using the default settings (8.1 s), and 51% using the modified settings (9.8 s). This example indicates the extent to which preliminary screening can be used to obtain tight bounds to the global optimal solution of the original MINLP problem, thus yielding a measure of confidence to the goodness of the solution.

IX. Concluding Remarks

We have presented an overview of algorithmic methods for process synthesis which rely on MINLP techniques. From this overall review, it is clear that considerable progress has been made over the last decade. Heuristics and thermodynamic targets, which have been dominant in the past, offer some valuable insights and motivate higher level representations for process synthesis. However, they do not provide a systematic framework for the modeling and decision making in process synthesis. In contrast, the mathematical programming approach offers a formalization and a basic modeling framework for simultaneous optimization in which trade-offs and interactions are systematically accounted for. Futhermore, the mathematical programming approach is more suitable for producing automated tools owing to the general mathematical representations that are used. At this time, we have reached a stage in which the modeling and solution of MINLP problems have become fact—especially with the increased computational power that is now available. Furthermore, the mathematical programming approach has increasingly been adopted by researchers in process systems over the last five years. This is not to say, of course, that MINLP optimization techniques are without their difficulties. Several of the outstanding problems that require attention for future work are the following:

(1) *Integrating approaches:* Clearly, a major question that remains is how to combine the heuristic search, mathematical programming and targeting approaches in such a way that the integration is conceptually consistent and rigorous, while yet exploiting the strengths of each approach. At the root of this problem lies the question of how to perform preliminary screening to eliminate alternatives, but within a rigorous multilevel design framework. In addition, the point below on aggregated models needs to be addressed to support this integration.

(2) *Developing superstructure representations at various levels of abstraction:* We should not view general design targets or aggregated models and fairly detailed process superstructures as different representations that are unrelated. Rather, they should be viewed as representations at different levels of abstraction in which the feasible space of the target model contains the feasible space of the superstructure. It should be interesting and useful to systematically aggregate detailed superstructure models, and vice-versa, to decompose target models that can be mapped to detailed flowsheets.

(3) *Generating novel superstructures for process flowsheets:* While for most homogeneous systems it is possible to postulate a general superstructure for analyzing the alternatives of interest, it is largely unclear how to systematically generate superstructures for process flowsheets involving different types of processing tasks or nonlinearities which give rise to infeasibilities that are

nontrivial to predict (e.g., azeotropic distillation). Furthermore, there is a need for considering representations that may lead to process intensification (e.g., integration of separation and reaction) and that may accommodate more readily the choice of different chemistries.

(4) *Finding new applications for process synthesis:* Most of the research work in synthesis has been aimed at conceptual flowsheets of continuous processes. There is a clear need to integrate the selection of chemical pathways in process synthesis (Douglas, 1990; Knight and McRae, 1993) as well as to consider the design of molecules (Maranas and Floudas, 1993). For example, the piping layout design (Guirardello and Swaney, 1993) has received limited attention, despite its importance. Also, work in the synthesis of batch processes is starting to emerge (Klossner and Rippin, 1984; Reklaitis, 1990; Papageorgaki and Reklaitis, 1990; Voudouris and Grossmann, 1993; Charalambides *et al.*, 1993).

(5) *Reducing combinatorial search:* In order to make mathematical programming tools useful for industrial problems, it is important that large combinatorial problems be handled effectively (this, of course, also includes large continuous models). The challenge here lies in accomplishing this goal without sacrificing optimality. New developments in MINLP algorithms are required for methods that can exploit more explicitly the logic and qualitative knowledge of a synthesis problen, making effective use of new ideas in polyhedral theory and new developments in advanced computer architectures.

(6) *Developing global optimization methods:* In most cases, the MINLP or NLP models for process synthesis involve nonconvexities in the continuous variables, which may give rise to several local or suboptimal optima. The difference between these solutions may be largely due to the expanded space of alternatives in these problems. Automating synthesis procedures that will sometimes produce poor solutions is obviously not satisfactory. Therefore, we need to develop global optimization methods that are both relevant to process synthesis problems and, preferably, rigorous in nature.

(7) *Handling of rigorous models:* In order to make synthesis techniques more applicable and relevant to industry, it is important that these techniques be either capable of explicitly handling complex process models or at least in some sense compatible with these models (e.g., in terms of bounding properties).

(8) *Synthesizing of process systems with multiple objectives:* Given the importance and potential impact of early design decisions in synthesis, it is highly desirable to develop computational frameworks that allow the evaluation and determination of trade-offs for a number of different attributes. Besides traditional economic measures, these include operability, safety, and environmental aspects.

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.

Acknowledgments

Most of the optimization developments for process synthesis described in this paper have been due to the following Ph.D. students at Carnegie Mellon: Soterios Papoulias, Marco Duran, Chris Floudas, Mark Shelton, Gary Kocis, Terry Yee, Ramesh Raman, Mark Daichendt, Ignacio Quesada and Metin Turkay. These efforts have been complemented by collaborations with Zdravko Kravanja, J. Viswanathan and Truls Gundersen. The author is grateful to Eastman Chemical Company and to the Engineering Design Research Center for financial support of this work.

References

Achenie, L. K. E., and Biegler, L. T. "Algorithmic Synthesis of Chemical Reactor Networks Using Mathematical Programming," *Ind. Eng. Chem. Fundam.* **25,** 621 (1986).

Achenie, L. K. E., and Biegler, L. T. "Developing Targets for the Performance Index of a Chemical Reactor Network: Isothermal Systems," *Ind. Eng. Chem. Res.* **27,** 1811 (1988).

Andrecovich, M. J., and Westerberg, A. W. "A Simple Synthesis Method Based on Utility Bounding for Heat-Integrated Distillation Sequences," *AIChE J.* **31,** 363 (1985a).

Andrecovich, M. J., and Westerberg, A. W. "An MILP Formulation for Heat-Integrated Distillation Sequence Synthesis," *AIChE J.* **31,** 1461 (1985b).

Bagajewicz, M. J., and Manousiouthakis, V. "Mass/Exchange Network Representation of Distillation Networks," *AIChE J.* **38**(11), 1769 (1992).

Balakrishna, S., and Biegler, L. T. "Constructive Targeting Approaches for the Synthesis of Chemical Reactor Networks," *Ind. Eng. Chem. Res.* **31,** 300 (1992a).

Balakrishna, S., and Biegler, L. T. "Targeting Strategies for the Synthesis and Energy Integration of Nonisothermal Reactor Networks," *Ind. Eng. Chem. Res.* **31,** 2152 (1992b).

Balas, E. "Disjunctive Programming and a Hierarchy of Relaxations for Discrete Optimization Problems," *SIAM J. Alg. Disc. Methods* **6,** 466–486 (1985).

Balas, E., Ceria, S., and Cornuejols, G. "A Lift-and-Project Cutting Plane Algorithm for Mixed 0–1 Programs," *Math. Program.* **58,** 295–324 (1993).

Beaumont, N. "An Algorithm for Disjunctive Programs," *Eur. J. Oper. Res.* **48,** 362–371 (1991).

Bolio, B., Daichendt, M., Iyer, R., Yee, T., and Grossmann, I. E. "SYNHEAT: An Interactive Program for Heat Exchanger Network Synthesis." Carnegie-Mellon University, Pittsburgh, PA, 1993.

Bolio, B., Quesada, I., and Grossmann, I. E. "GLOBESEP: An Interactive Program for the Global Optimization of Separation Networks." Carnegie-Mellon University, Pittsburgh, PA, 1994.

Borchers, B., and Mitchell, J. E. "An Improved Branch and Bound Algorithm for Mixed Integer Nonlinear Programs," TIMS/ORSA Meeting (1992).

Bremicker, J. F., Papalambros, P. Y., and Loh, H. T. "Solution of Mixed-Discrete Structural Optimization with a New Sequential Linearization Model," *Comput. Struct.* **37,** 451–461 (1990).

Brooke, A., Kendrick, D., and Meeraus, A. "GAMS—A User's Guide." Scientific Press, Palo Alto, CA, 1988.

Carlberg, N. A., and Westerberg, A. W. "Temperature-Heat Diagrams for Complex Columns. 2. Method for Side Strippers and Enrichers," *Ind. Eng. Chem. Res.* **28,** 1379 (1989a).

Carlberg, N. A., and Westerberg, A. W. "Temperature-Heat Diagrams for Complex Columns. 3. Underwood's Method for the Petlyuk Configuration," *Ind. Eng. Chem. Res.* **28,** 1386 (1989b).

Cavalier, T. M., and Soyster, A. L. "Logical Deduction via Linear Programming," IMSE Working Paper 87-147. Department of Industrial and Management Systems Engineering, Pennsylvania State University, University Park, 1987.

Cavalier, T. M., Pardalos, P. M., and Soyster, A. L. "Modelling and Integer Programming Techniques applied to Propositional Calculus," *Comput. Oper. Res.* **17**(6), 561–570 (1990).

Cerda, J., and Westerberg, A. W. "Synthesizing Heat Exchanger Networks Having Restricted Stream/Stream Matches Using Transportation Problem Formulations," *Chem. Eng. Sci.* **38,** 1723 (1983).

Charalambides, M. S., Shah, N., and Pantelides, C. C. "Optimal Batch Process Synthesis," Paper No. 153c, AIChE Meeting, St. Louis, MO (1993).

Chitra, S. P., and Govind, R. "Synthesis of Optimal Serial Reactor Structures for Homogeneous Reactions. Part I: Isothermal Reactors," *AIChE J.* **31,** 177 (1985a).

Chitra, S. P., and Govind, R. "Synthesis of Optimal Serial Reactor Structures for Homogeneous Reactions. Part II: Nonisothermal Reactors," *AIChE J.* **31,** 185 (1985b).

Ciric, A. R., and Floudas, C. A. "Heat Exchanger Network Synthesis without Decomposition," *Comput. Chem. Eng.* **15,** 385–396 (1991).

Clocksin, W. F., and Mellish, C. S., "Programming in Prolog," Springer–Verlarg, Heidelberg, 1981.

Colberg, R. D., and Morari, M. "Area and Capital Cost Targets for Heat Exchanger Network Synthesis with Constrained Matches and Unequal Heat Transfer Coefficients," *Comput. Chem. Eng.* **14,** 1 (1990).

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

Crowder, H., Johnson, E. L., and Padberg, M. "Solving Large-Scale Zero-One Linear Programming Problems," *Oper. Res.* **31,** 803–834 (1983).

Daichendt, M. M., and Grossmann, I. E. "Preliminary Screening for the MINLP Synthesis of Process Systems. I. Aggregation and Decomposition Techniques," *Comput. Chem. Eng.* **18,** 663 (1994a).

Daichendt, M. M., and Grossmann, I. E. "Preliminary Screening for the MINLP Synthesis of Process Systems. II. Heat Exchanger Networks," *Comput. Chem. Eng.* **18,** 679 (1994b).

Dhole, V. R., and Linnhoff, B. "Distillation Column Targets," *Proc. Eur. Symp. Comput. Aided Process Design (Escape-1)* (1992).

Ding, M., and Sargent, R. W. H. "A Combined SQP and Branch and Bound Algorithm for MINLP Optimization," Internal Report, Centre for Process Systems Engineering, Imperial Colleges London, 1992.

Doherty, M. F., and Cardarola, G. A. "Design and Synthesis of Homogeneous Azeotropic Distillations. 3. The Sequencing of Columnds for Azeotropic and Extractive Distillation," *Ind. Eng. Chem. Fundam.* **24,** 474–485 (1985).

Douglas, J. M. "A Hierarchical Decision Procedure for Process Synthesis," *AIChE J.* **31,** 353 (1985).

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

Douglas, J. M. "Synthesis of Multistep Reaction Processes," *in* "Foundations of Computer-Aided Design" (J. J. Siirola, I. E. Grossmann, and G. Stephanopoulos, eds.). Cache-Elsevier, Amsterdam, 1990.

Duran, M. A., and Grossmann, I. E. "Simultaneous Optimization and Heat Integration of Chemical Processes," *AIChE J.* **32,** 123 (1986a).

Duran, M. A., and Grossmann, I. E. "A Mixed-Integer Nonlinear Programming Algorithm for Process Systems Synthesis," *AIChE J.* **32,** 592 (1986b).

Duran, M. A., and Grossmann, I. E. "An Outer-Approximation Algorithm for a Class of Mixed-integer Nonlinear Programs," *Math. Program.* **36,** 307 (1986c).

El-Halwagi, M., and Manousiouthakis, V. "Synthesis of Mass Exchange Networks," *AIChE J.* **35**(8), 1233 (1989a).

El-Halwagi, M., and Manousiouthakis, V. "Design and Analysis of Mass Exchange Networks with Multicomponent Targets," Paper 137f, AIChE Meeting San Francisco (1989b).

El-Halwagi, M., and Manousiouthakis, V. "Automatic Synthesis of Mass-Exchanger Networks with Single Component Targets," *Chem. Eng. Sci.* **45**(9), 2813 (1990).

Eliceche, A. M., and Sargent, R. W. H. "Synthesis and Design of Distillation Sequences," *Inst. Chem. Eng. Symp. Ser.* **61,** 1–22 (1986).

Fletcher, R., and Leyffer, S. "Solving Mixed Integer Nonlinear Programs by Outer Approximation," *Math. Program.* **66,** 327 (1994).

Floquet, P., Pibouleau, L., and Domenach, S. "Mathematical Programming Tools for Chemical Engineering Process Design Synthesis," *Chem. Eng. Process.* **23,** 1 (1988).

Floudas, C. A. "Separation Synthesis of Multicomponent Feed Streams into Multicomponent Product Stream," *AIChE J.* **33,** 540–550 (1987).

Floudas, C. A., and Aggarwal, A. "A Decomposition Strategy for Global Optimum Search in the Pooling Problem," *ORSA J. Comput.* **2,** 225–235 (1990).

Floudas, C. A., and Grossmann, I. E. "Algorithmic Approaches to Process Synthesis: Logic and Global Optimization," *in* "Foundations of Computer-Aided Process Design" (M. F. Doherty and L. T. Biegler eds.). Snowmass, CO, 1994.

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

Floudas, C. A., and Visweswaran, V. "A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLPs-I Theory," *Comput. Chem. Eng.* **4,** 1397–1417 (1990).

Floudas, C. A., Ciric, A. R., and Grossmann, I. E. "Automatic Synthesis of Optimum Heat Exchanger Network Configurations," *AIChE J.* **32,** 276 (1986).

Fonyó, Z. "Thermodynamic Analysis of Rectification. I. Reversible Model of Rectification," *Int. Chem. Eng.* **14,** 18 (1974a).

Fonyó, Z. "Thermodynamic Analysis of Rectification. II. Finite Cascade Models," *Int. Chem. Eng.* **14,** 203 (1974b).

Fonyó, Z., and Mizsey, P. "A Global Approach to the Synthesis and Preliminary Design of Integrated Total Flowsheets," Annual AIChE Meeting, Chicago (1990).

Friedler, F., Tarjan, K., Huang, Y. W., and Fan, L. T. "An Accelerated Branch and Bound Method for Process Synthesis," presented at the 4th World Congress of Chemical Engineering, Karlsruhe (1991).

Friedler, F., Tarjan, K., Huang, Y. W., and Fan, L. T. "Graph Theoretic Approach to Process Synthesis: Axioms and Theorems," *Chem. Eng. Sci.* **47,** 1973–1988 (1992).

Friedler, F., Tarjan, K., Huang, Y. W., and Fan, L. T. "Graph Theoretic Approach to Process Synthesis: Polynomial Algorithm for Maximal Structure Generation," *Comput. Chem. Eng.* **17,** 929–942 (1993).

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

Glasser, D., Hildebrandt, D., and Crowe, C. "A Geometric Approach to Steady Flow Reactors: The Attainable Region and Optimization in Concentration Space," *Ind. Eng. Chem. Res.* **26,** 1803 (1987).

Glavic, P., Kravanja, Z., and Homsak, M. "Heat Integration of Reactors. I. Criteria for Placement of Reactors into the Flowsheet," *Chem. Eng. Sci.* **43,** 593 (1988).

Gomez-Muñoz, A., and Seader, J. D. "Synthesis of Distillation Trains by Thermodynamic Analysis," *Comput. Chem. Eng.* **9,** 311 (1985).

Grossmann, I. E. "Mixed-Integer Programming Approach for the Synthesis of Integrated Process Flowsheets," *Comput. Chem. Eng.* **9,** 463 (1985).

Grossmann, I. E. "MINLP Optimization Strategies and Algorithms for Process Synthesis," *in* "Foundations of Computer-Aided Design" (J. J. Siirola, I. E. Grossmann, and G. Stephanopoulos, eds.). Cache-Elsevier, Amsterdam, 1990a.

Grossmann, I. E. "Mixed-Integer Nonlinear Programming Techniques for the Synthesis of Engineering Systems," *Res. Eng. Des.* **1,** 205 (1990b).

Grossmann, I. E., and Daichendt, M. M. "New Trends in Optimization-based Approaches for Process Synthesis," *in* Proceedings of Process Systems Engineering, Korea (1994).

Guirardello, R., and Swaney, R. E. "Process Piping Layout Design and Optimization," Paper No. 151g, AIChE Meeting, St. Louis, MO (1993).

Gundersen, T., and Grossmann, I. E. "Improved Optimization Strategies for Automated Heat Exchanger Network Synthesis through Physical Insights," *Comput. Chem. Eng.* **14,** 925 (1990).

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

Gupta, O. K., and Ravindran, V. "Branch and Bound Experiments in Convex Nonlinear Integer Programming," *Manage. Sci.* **31**(12), 1533–1546 (1985).

Hendry, J. E., and Hughes, R. R. "Generating Separation Process Flowsheets," *Chem. Eng. Prog.* **68,** 69 (1972).

Hendry, J. E., Rudd, D. F., and Seader, J. D. "Synthesis in the Design of Chemical Processes," *AIChE J.* **19,** 1 (1973).

Hildebrandt, D., Glasser D., and Crowe, C. "Geometry of the Attainable Region Generated by Reaction and Mixing: With and without Constraints," *Ind. Eng. Chem. Res.* **29,** 49 (1990).

Hlavacek, V. "Synthesis in the Design of Chemical Processes," *Comput. Chem. Eng.* **2,** 67–75 (1978).

Horst, R. "Deterministic Method in Constrained Global Optimization: Some Recent Advances and Fields of Application," *Nav. Res. Logistics* **37,** 433–471 (1990).

Horst, R., and Tuy, T. "Global Optimization: Deterministic Approaches." Springer-Verlag, Berlin and New York, 1990.

Ichikawa, A., and Fan, L. T. "Optimal Synthesis of Process Systems," Necessary Condition for Optimal System and Its Use in Synthesis of Systems," *Chem. Eng. Sci.* **28,** 357 (1973).

Kakhu, A. I., and Flower, J.R. "Synthesising Heat-Integrated Distillation Sequences Using Mixed Integer Programming," *Chem. Eng. Res. Des.* **66,** 241 (1988).

Kelley, J. E., Jr., 'The Cutting-Plane Method for Solving Convex Programs," *J. SIAM* **8,** 703–712 (1960).

Kirkwood, R. L., Locke, M. H., and Douglas, J. M. "A Prototype Expert System for Synthesizing Chemical Process Flowsheets," *Comput. Chem. Eng.* **12,** 329 (1988).

Klossner, J., and Rippin, D. W. T. "Combinatorial Problems in the Design of Multiproduct Batch Plants-Extensions to Multiplant and Partly Parallel Operations," AIChE Meeting, San Francisco (1984).

Knight, J. R., and Doherty, M. F. "Optimal Design and Synthesis of Homogeneous Azeotropic Distillation Sequences," *Ind. Eng. Chem. Res.* **28,** 564–572 (1989).

Knight, J. R., and McRae, G. J. "An Approach to Process Integration based on the Choice of the System Chemistry," Paper No. 153d, AIChE Meeting, St. Louis, MO (1993).

Kocis, G. R., and Grossmann, I. E. "Relaxation Strategy for the Structural Optimization of Process Flow Sheets," *Ind. Eng. Chem. Res.* **26,** 1869 (1987).

Kocis, G. R., and Grossmann, I. E. "Global Optimization of Nonconvex Mixed-Integer Nonlinear Programming (MINLP) Problems in Process Synthesis," *Ind. Eng. Chem. Res.* **27,** 1407 (1988).

Kocis, G. R., and Grossmann, I. E. "Computational Experience with DICOPT Solving MINLP Problems in Process Systems Engineering," *Comput. Chem. Eng.* **13,** 307 (1989a).

Kocis, G. R., and Grossmann, I. E. "A Modelling and Decomposition Strategy for the MINLP Optimization of Process Flowsheets," *Comput. Chem. Eng.* **13,** 797 (1989b).

Koehler, J., Aguirre, P., and Blass, E. "Evolutionary Thermodynamic Synthesis of Zeotropic Distillation Sequences," *Gas Separ. Purif.* **6,** 4153 (1992).

Kokossis, A. C., and Floudas, C. A. "Synthesis of Isothermal Reactor-Separator-Recycle Systems," *Chem. Eng. Sci.* **46,** 1361–1383 (1991).

Kravanja, Z., and Grossmann, I. E. "PROSYN, An MINLP Process Synthesizer," *Comput. Chem. Eng.* **14,** 1363 (1990).

Kravanja, Z., and Grossmann, I. E. "Recent Developments in PROSYN—A Topology MINLP Synthesizer," *Comput. Chem. Eng.* (1994).

Lang, Y. D., Biegler, L. T., and Grossmann, I. E. "Simultaneous Optimization and Heat Integration with Process Simulators," *Comput. Chem. Eng.* **12,** 311 (1988).

Laroche, L., Bekiaris, N., Andersen, H. W., and Morari, M. "Homogeneous Azotropic Distillation: Separability and Flowsheet Synthesis," *Ind. Eng. Chem. Res.* **31,** 2190–2209 (1992).

Lee, K. F., Masso, A. H., and Rudd, D. F. "Branch and Bound Synthesis of Integrated Process Designs," *Ind. Eng. Chem. Fundam.* **9,** 48 (1970).

Leyffer, S. "Deterministic Methods for Mixed-Integer Nonlinear Programming," Ph.D. Thesis, Department of Mathematics and Computer Science, University of Dundee, Dundee (1993).

Linnhoff, B. "Pinch Analysis—A State of the Art Overview," *Trans. Inst. Chem. Eng.* **71**(a), 503–522 (1993).

Linnhoff, B., and Hindmarsh, E. "The Pinch Design Method of Heat Exchanger Networks," *Chem. Eng. Sci.* **38,** 745 (1983).

Linnhoff, B. et al., "User Guide on Process Integration for the Efficient Use of Energy." Inst. Chem. Eng., Rugby, 1982.

Magnanti, T. L., and Wong, R. T. "Acclerated Benders Decomposition: Algorithm Enhancement and Model Selection Criteria," *Oper. Res.* **29,** 464–484 (1981).

Mahalec, V., and Motard, R. L. "Procedures for the Initial Design of Chemical Processing Systems," *Comput. Chem. Eng.* **1,** 57 (1977a).

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

Manousiouthakis, V., and Sourlas, D. "A Global Optimization Approach to Rationally Constrained Rational Programming," *Chem. Eng. Commun.* **115,** 127–147 (1992).

Maranas, C. D., and Floudas, C. A. "Global Minimum Potential Energy Conformation of Small Molecules," Paper No. 151d, AIChE Meeting, St. Louis, MO (1993).

Masso, A. H., and Rudd, R. D. "The Synthesis of System Designs. II: Heuristic Structuring," *AIChE J.* **15,** 10 (1969).

McCormick, G. P. "Computability of Global Solutions to Factorable Nonconvex Programs: Part I. Convex Underestimating Problems," *Math. Program.* **10,** 146–175 (1976).

Nabar, S. and Schrage, L. "Modelling and Solving Nonlinear Integer Programming Problems," presented at Annual AIChE Meeting, Chicago (1991).

Nath, R., and Motard, R. L. "Evolutionary Synthesis of Separation Processes," *AIChE J.* **27,** 578 (1981).

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

Nishida, N., Stephanopoulos, G., and Westerberg, A. W. "A Review of Process Synthesis," *AIChE J.* **27,** 321 (1981).

Omtveit, T., and Lien, K. "Reactor System Design Revisited," Paper No. 153b, AIChE Meeting, St. Louis, MO (1993).

Papageorgaki, S., and Reklaitis, G. V. "Optimal Design of Multipurpose Batch Plants. I. Problem Formulation," *Ind. Eng. Chem. Res.* **29,** 2054 (1990).

Papalexandri, K. P., Pistikopoulos, E. N., and Floudas, C. A. "Mass Exchange Networks for Waste Minimization: A Simultaneous Approach," Paper No. 153e, AIChE Meeting, St. Louis, MO (1993).

Papoulias, S. A., and Grossmann, I. E. "A Structural Optimization Approach in Process Synthesis. Part I. Utility Systems," *Comput. Chem. Eng.* **7,** 695 (1983a).

Papoulias, S. A., and Grossmann, I. E. "A Structural Optimization Approach in Process Synthesis. Part II. Heat Recovery Networks," *Comput. Chem. Eng.* **7,** 707 (1983b).

Papoulias, S. A., and Grossmann, I. E. "A Structural Optimization Approach in Process Synthesis. Part III. Total Processing Systems," *Comput. Chem. Eng.* **7,** 723 (1983c).

Petlyuk, F. B., Platonov, V. M., and Slavinski, D. M. "Thermodynamically Optimal Method for Separating Multicomponent Mixtures," *Int. Chem. Eng.* **5,** 555 (1965).

Pinto, J. M., and Grossmann, J. E., "Optimal Cyclic Scheduling of Multistage Continuous Multiproduct Plants," *Comput Chem. Eng.* **18,** 797–816 (1994).

Powers, G. J., "Heuristic Synthesis in Process Development," *Chem. Eng. Prog.* **68,** 88 (1972).

Quesada, I., and Grossmann, I. E. "An LP/NLP Based Branch and Bound Algorithm for Convex MINLP Optimization Problems," *Comput. Chem. Eng.* **16,** 937–947 (1992).

Quesada, I., and Grossmann, I. E. "Global Optimization Algorithm for Heat Exchanger Networks," *Ind. Eng. Chem. Res.* **32,** 487 (1993).

Quesada, I., and Grossmann, I. E. "Global Optimization Algorithm of Process Networks with Multicomponent Flows," *Comput. Chem. Eng.* **19,** 1219–1242 (1995a).

Quesada, I., and Grossman, I. E. *J. Global Optim.* **6,** 39–76 (1995b).

Raman, R., and Grossmann, I. E. "Relation Between MILP Modelling and Logical Inference for Chemical Process Synthesis," *Comput. Chem. Eng.* **15,** 73 (1991).

Raman, R., and Grossmann, I. E. "Symbolic Integration of Logic in Mixed Integer Linear Programming Techniques for Process Synthesis," *Comput. Chem. Eng.* **17,** 909 (1993).

Raman, R., and Grossmann, I. E. "Modelling and Computational Techniques for Logic Based Integer Programming," *Comput. Chem. Eng.* **18,** 563–578 (1994).

Rathore, R. N. S., van Wormer, K. A., and Powers, G. J. "Synthesis Strategies for Multicomponent Separation Systems with Energy Integration," *AIChE J.* **20,** 491 (1974a).

Rathore, R. N. S., van Wormer, K. A., and Powers, G. J. "Synthesis of Distillation Systems with Energy Integration," *AIChE J.* **20,** 940 (1974b).

Reklaitis, G. V. "Progress and Issues in Computer-Aided Batch Process Design," *in* "Foundations of Computer-Aided Design" (J. J. Siirola, I. E. Grossmann, and G. Stephanopoulos, eds.). Cache-Elsevier, Amsterdam, 1990.

Rigg, T. J. "Synthesis of Separation Systems," A Preliminary Report for the Degree of Doctor of Philosophy in Chemical Engineering, University of Wisconsin-Madison (1991).

Rippin, D. W. T. "Introduction: Approaches to Chemical Process Synthesis," *in* "Foundations of Computer-Aided Design" (J. J. Siirola, I. E. Grossmann, and G. Stephanopoulos, eds.). Cache-Elsevier, Amsterdam 1990.

Rudd, D. F., "The Synthesis of System Designs. I. Elementary Decomposition Strategy," *AIChE J.* **14,** 343 (1968).

Rudd, D. F., and Watson, C . C. "Strategy of Process Engineering." Wiley, New York, 1968.

Sahinidis, N. V. "Accelerating Branch and Bound in Continuous Global Optimization," Paper MA 36.2, TIMS/ORSA Meeting, Phoenix, AZ (1993).

Sahinidis, N. V., and Grossmann, I. E. "MINLP Model for Cyclic Multiproduct Scheduling on Continuous Parallel Lines," *Comput. Chem. Eng.* **15,** 85 (1991a).

Sahinidis, N. V., and Grossmann, I. E. "Convergence Properties of Generalized Benders Decomposition," *Comput. Chem. Eng.* **15,** 481 (1991b).

Sargent, R. W. H., and Gaminibandara, K. "Introduction: Approaches to Chemical Process Synthesis," *in* "Optimization in Action" (L. C. W. Dixon, ed.). Academic Press, London, 1976.

Schrijver, A. "Theory of Linear and Integer Programming." Wiley, New York, 1986.

Seader, J. D., and Westerberg, A. W. "A Combined Heuristic and Evolutionary Strategy for Synthesis of Simple Separation Sequences," *AIChE J.* **23,** 951 (1977).

Shelton, M. R., and Grossmann, I. E. "Optimal Synthesis of Integrated Refrigeration Systems. I: Mixed-integer Programming Model," *Comput. Chem. Eng.* **10,** 445 (1986a).

Shelton, M. R., and Grossmann, I. E. "Optimal Synthesis of Integrated Refrigeration Systems. II: Implicit Enumeration Algorithm," *Comput. Chem. Eng.* **10,** 461 (1986b).

Siirola, J. J., and Rudd, D. F. "Computer-Aided Synthesis of Chemical Process Designs," *Ind. Eng. Chem. Fundam.* **10,** 353 (1971).

Siirola, J. J., Powers, G. J., and Rudd, D. F. "Synthesis of System Designs, III: Toward a Process Concept Generator," *AIChE J.* **17,** 677 (1971).

Smith, R., and Linnhoff, B. "The Design of Separators in the Context of Overall Processes," *Chem. Eng. Res. Des.* **66,** 195 (1988).

Stephanopoulos, G., and Westerberg, A. W. "Studies in Process Synthesis, II. Evolutionary Synthesis of Optimal Process Flowsheets," *Chem. Eng. Sci.* **31,** 195 (1976).

Stichlmair, J., Fair, J., and Bravo, J. L. "Separation of Azeotropic Mixtures via Enhanced Distillation," *Chem. Eng. Prog.* **85,** 63–69 (1989).

Swaney, R. E. "Global Solution of Algebraic Nonlinear Programs," Paper No.22f, AIChE Meeting, (1990).

Terranova, B. E., and Westerberg, A. W. "Temperature-Heat Diagrams for Complex Columns. 1. Intercooled/Interheated Distillation Columns," *Ind. Eng. Chem. Res.* **28,** 1374 (1989).

Turkay, M., and Grossmann, I.E. "A Logic Based Outer-Approximation Algorithm for MINLP Optimization of Process Flowsheets," AIChE Annual Meeting, San Francisco (1994).

Umeda, T., Hirai, A., and Ichikawa, A. "Synthesis of Optimal Processing System by an Integrated Approach," *Chem. Eng. Sci.* **27,** 795 (1972).

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

Van Dongen, D. B., and Doherty, M. F. "Design and Synthesis of Homogeneous Azeotropic Distillation. 1. Problem Formulation for Single Column," *Ind. Eng. Chem. Fundam.* **24,** 454 (1985).

Van Roy, T. J., and Wolsey, L. A. "Solving Mixed Integer Programs by Automatic Reformulation," *Oper. Res.* **35,** 45–57 (1987).

Viswanathan, J., and Grossmann, I. E. "A Combined Penalty Function and Outer-Approximation Method for MINLP Optimization," *Comput. Chem. Eng.* **14,** 769 (1990).

Viswanathan, J., and Grossmann, I. E. "Optimal Feed Locations and Number of Trays for Distillation Columns with Multiple Feeds," *Ind Eng. Chem. Res.* **32,** 2942–2949 (1993).

Visweswaran, C., and Floudas, C. A. "A Global Optimization Algorithm (GOP) for Certain Classes of Nonconvex NLPs. II. Application of Theory and Test Problems," *Comput. Chem. Eng.* **14**(2), 1419–1434 (1990).

Voudouris, V. T., and Grossmann, I. E. "Optimal Synthesis of Multiproduct Batch Plants with Cyclic Scheduling and Inventory Considerations," *Ind. Eng. Chem. Res.* **32,** 1962 (1993).

Wahnschafft, O. M., Jurain, T. P., and Westerberg, A. W. "SPLIT: A Separation Process Designer," *Comput. Chem. Eng.* **15,** 565–581 (1991).

Wahnschafft, O. M., LeRudulier, J. P., Blania, P., and Westerberg, A. W. "SPLIT: II. Automated Synthesis of Hybrid Liquid Separation Systems," *Comput. Chem. Eng.* **16,** S305–S312 (1992).

Wahnschafft, O. M., Wareck, J. S., and Ahmad, S. "Distillation System Synthesis Based on Thermodynamic Targeting," Paper No. 152f, AIChE Meeting, St. Louis, MO (1993).

Wehe, R. R., and Westerberg, A. W. "An Algorithmic Procedure for the Synthesis of Distillation Sequences with Bypass," *Comput. Chem. Eng.* **11,** 619–627 (1987).

Westerberg, A. W. "The Synthesis of Distillation Based Separation," *Comput. Chem. Eng.* **9,** 421 (1985).

Westerlund, T., and Pettersson, F. "A Cutting Plane Method for Solving Convex MINLP Problems," Report 92-124-A. Process Design Laboratory, Abo Akademi, 1992.

Williams, H. P. "Model Building in Mathematical Programming," Wiley, Chichester, 1988.

Yee, T. F., Grossmann, I. E., and Kravanja, Z. "Simultaneous Optimization Models for Heat Integration. I. Energy and Area Targeting," *Comput. Chem. Eng.* **14,** 1151 (1990a).

Yee, T. F., and Grossmann, I. E. "Simultaneous Optimization Models for Heat Integration. II: Heat Exchanger Network Synthesis," *Comput. Chem. Eng.* **14,** 1165 (1990b).

Yee, T. F., Grossmann, I. E., and Kravanja, Z. "Simultaneous Optimization Models for Heat Integration. III. Optimization of Process Flowsheets and Heat Exchanger Networks," *Comput. Chem. Eng.* **14,** 1185 (1990c).

Yuan, X., Zhang, S., Pibouleau, L., and Domenech, S. "Une Méthode d'optimisation Nonlinéare en Variables pour la Conception de Procèdés," *Oper. Res.* **22,** 331 (1988).