Journal of Global Optimization **22:** 59–95, 2002.

Deterministic global optimization in isothermal reactor network synthesis

W. R. ESPOSITO[1,2] and C. A. FLOUDAS[1,*]
[1]*Department of Chemical Engineering, Princeton University, Princeton, NJ 08544-5263, USA (E-mail: floudas@titan.princeton.edu)*
[2]*Current address: Praxair, Inc., 175 East Park Drive, Tonawanda, NY 14150, USA *Corresponding author*

Abstract. The reactor network synthesis problem involves the simultaneous determination of the structure and operating conditions of a reactor system to optimize a given performance measure. This performance measure may be the yield of a given product, the selectivity between products, or the overall profitability of the process. The problem is formulated as a nonlinear program (NLP) using a superstructure based method in which plug flow reactors (PFRs) in the structure are modeled using differential-algebraic equations (DAEs). This formulation exhibits multiple local minima. To overcome this, a novel deterministic global optimization method tailored to the special structure and characteristics of this problem will be presented. Examples of isothermal networks will be discussed to show the nature of the local minima and illustrate various components of the proposed approach.

Key words: Reactor Network Synthesis, Differential-Algebraic Equations, Deterministic Global Optimization

1. Introduction

The reactor synthesis problem can be simply stated as: given a set of reactions and feeds, what is the configuration and operational characteristics of a reactor system which will optimize a given performance measure? This performance measure can range from the yield of a product, to a complicated measure involving both capital and operational economics. As a result of the influence the reactor section has on the operation of a complete process, the determination of a true optimal configuration is needed in order to maximize the overall profitability.

The solution of this problem has been approached in many different ways. Early work was directed toward choosing the optimal mixing within a reactor for a given reaction system (Dyson and Horn, 1967; Horn and Tsai, 1967). Other work involved determining the optimal configuration for a given 'type' of reaction scheme. Many examples of this appear in the book by Levenspiel (1993) and the work of (Chitra and Govind, 1981, 1985). Other additional literature related to these approaches is discussed in the references.

More recently, two different methods, the superstructure based and targeting based, have emerged as the predominant approaches for the solution to this problem. Hildebrandt and Biegler (1995) provided an overview of these two methods,

but focused on the targeting based approaches. In these approaches, the concepts of reaction and mixing are used to generate a target, or the maximum achievable performance. Using this information, a physical reactor network is then determined to meet this measure. Horn (1964) presented a geometric based method for determining the 'attainable region' in concentration space of a given reaction scheme. The works of Glasser et al. (1987), Hildebrandt et al. (1990), Hildebrandt and Glasser (1990) and Feinberg and Hildebrandt (1997) discussed the mathematical properties of the attainable region, and extended the work to include complicating parts, such as temperature effects and reactor constraints. The recent works of Feinberg (1999, 2000a, b) have farther explored the properties of various reactor types which make up the boundary of the attainable region. The geometrically based methods have the drawback that extensions passed two or three dimensions are quite difficult. In order to overcome this, optimization based methods as opposed to geometric approaches have been used to determine the boundary of the attainable region. Balakrishna and Biegler (1996) presented an overview of these optimization based targeting methods. The works of Achenie and Biegler (1988) and Balakrishna and Biegler (1992a, b) also provided examples of targeting approaches for both isothermal and non-isothermal reaction systems.

The second class of approaches are the 'superstructure based optimization' methods. In these approaches, first a superstructure of possible reactor configurations is postulated. The structure is then formulated into an optimization problem and solved using various methods. Ong (1986) optimized a serial CSTR configuration using Dynamic Programming. Achenie and Biegler (1986, 1990) solved a superstructure based on constant-dispersion reactors and recycle reactors respectively, using a two-point boundary value formulation popular in the optimal control literature. Kokossis and Floudas (1990, 1994a) postulated a novel superstructure containing simple CSTRS and PFRs (which are approximated using a cascade of CSTRs), and generated a mixed-integer nonlinear program (MINLP). This is done for the cases of isothermal and non-isothermal reactors, respectively. The work was extended to include reactor stability considerations (Kokossis and Floudas, 1994b), and reactor-separator-recycle systems (Kokossis and Floudas, 1991). Schweiger and Floudas (1999) proposed a similar approach, but did not include integer variables, nor any approximation of the PFRs in the system. This results in an optimal control formulation which is solved using a sequential method. Cordero et al. (1997) presented an approach using simulated annealing to solve an MINLP formulation.

Some researchers have developed methods which are a combination of the targeting and superstructure approaches. Lakshmanan and Biegler (1996) used the basic concepts of the attainable-region to postulate a concise superstructure for a given reaction scheme. The problem is then formulated as an mixed-integer optimal control problem and solved by reducing it to an MINLP using orthogonal collocation to approximate the PFR dynamics. Marcoulaki and Kokossis (1999) presented a similar type of approach without using an MINLP formulation. A simulated

annealing method generates a reactor configuration using targeting ideas. Then the configuration is evaluated using a simulation technique. The process is then repeated until an 'optimal' configuration and operating conditions are determined.

A common limitation of all the optimization based approaches is the nature of the resulting formulation. Due to the bilinearities which result from the mass balances, and the arbitrary and often nonlinear kinetics, the formulations are non-convex. In almost all cases this leads to the existence of multiple local minima. Many of the authors in the previous references have made note of this (Achenie and Biegler, 1986; Kokossis and Floudas, 1990; Hildebrandt and Biegler, 1995; Schweiger and Floudas, 1999). In fact, some published solutions have later been shown to be local in nature (Schweiger and Floudas, 1999). The application of simulated annealing can only increase the chance of determining a global solution, but a local minimum is typically obtained. A method which can guarantee the determination of a global solution does not currently exist.

In this paper, a novel deterministic global optimization method will be presented to solve the isothermal reactor network synthesis problem. The synthesis problem will be framed using a superstructure based approach, with the resulting optimal control formulation solved locally using standard NLP techniques (similar to the approach presented by (Schweiger and Floudas, 1999)). The global optimization method is based on the αBB (Adjiman et al. 1998a, b), a branch-and-bound approach originally developed for twice continuously differentiable NLPs. The approach was extended to handle optimal control formulations by Esposito and Floudas (2000a). Section 2 will present the basic formulation of the problem, with various simplifications and reformulations which make the application of a global optimization method easier. Section 3 will discuss the basic concepts of the αBB and the various customizations for the unique nature of this formulation. Finally, in Section 4 examples of isothermal networks will be presented to illustrate the various aspects of the approach.

2. Problem Formulation

The reactor network synthesis problem will be formulated using a superstructure based framework. Only isothermal operation under the assumption of constant density will be considered. The following information is assumed to be known: (1) the composition of the inlet stream, (2) the reaction mechanism and rate constants, and (3) the performance index to be optimize. The following information will be obtained from the optimization procedure: (1) the overall structure of the reactor network, (2) the relative flowrates and concentrations of each stream in the network, and (3) the residence time of all reactors in the system.

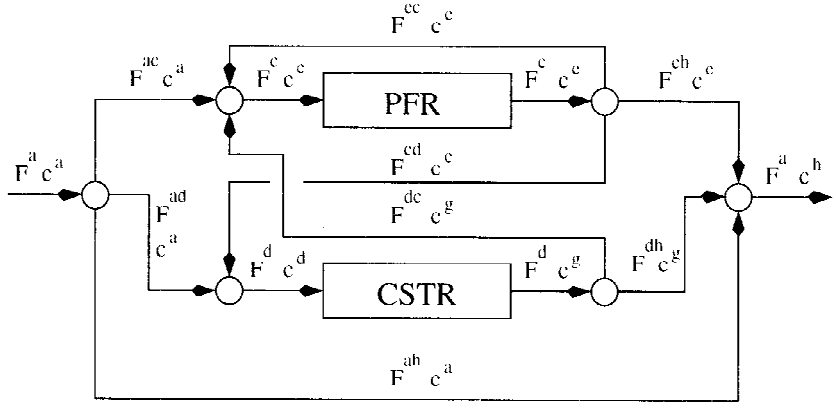


Figure 1. Reactor network superstructure.

2.1. REACTOR SUPERSTRUCTURE

The key points in the development of a reactor superstructure are provided by Schweiger and Floudas (1999). For this study, a superstructure containing one PFR with recycle, one CSTR, and a system bypass will be considered. Within the superstructure, flow splitters are located at the inlet to the network and the outlet of each of the reactors. Flow mixers are located at the inlet to each of the reactors, and at the outlet of the network. This superstructure is illustrated in Figure 1. The definition of the variables within this superstructure are given in Table 1. It should be noted that the constant density approximation is directly included in the superstructure, for instance, the volumetric flow into and out of a reactor is treated as the same variable. All flowrates, $\boldsymbol{F}$, are measured in l/time, and the concentrations, $\boldsymbol{c}$, in mol/l.

2.2. MATHEMATICAL FORMULATION

Using the superstructure depicted in Figure 1, its mathematical formulation is detailed in the following sections.

2.2.1. *Given Quantities*

The following quantities are given in the definition of the problem: (1) $\boldsymbol{c^a}$, the inlet concentration of all components, (2) $\boldsymbol{k}$, the set of reaction rate constants for the given kinetic model, (3) $\boldsymbol{\nu}$, the matrix of stoichiometric coefficients which describes the given kinetic model, and (4) reasonable bounds on the size of the reactors. The inlet flowrate, F^a, is set equal to 1 L/time (time units are not important). This sets all the flowrates in the system to be relative to the flowrate of the inlet stream.

Table 1. Definitions of variables within the superstructure.

F^a	Network feed and product flowrate
F^{ac}	Flowrate from the feed splitter to the PFR mixer
F^{ad}	Flowrate from the feed splitter to the CSTR mixer
F^{ah}	System bypass flowrate
F^c	Flowrate through the PFR.
F^{ch}	Flowrate from the PFR splitter to the product mixer.
F^{cd}	Flowrate from the PFR splitter to the CSTR mixer.
F^{ec}	PFR recycle flowrate.
F^d	Flowrate through the CSTR.
F^{dc}	Flowrate from the CSTR splitter to the PFR mixer
F^{dh}	Flowrate from the CSTR splitter to the product mixer
$\boldsymbol{c}^a$	Network feed concentrations
$\boldsymbol{c}^c$	PFR inlet concentrations
$\boldsymbol{c}^e$	PFR outlet concentrations
$\boldsymbol{c}^d$	CSTR inlet concentrations
$\boldsymbol{c}^g$	CSTR outlet concentrations
$\boldsymbol{c}^h$	Network product concentrations
τ^m	Residence time of the CSTR
τ^t	Residence time of the PFR

2.2.2. *Performance Measure*

The objective of the problem is to optimize some performance measure based on the reactor network. The objective function will be written as a minimization for the sake of generality:

$$\min_{\boldsymbol{x}} g(\boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x}$ is the full set of algebraic variables, including sizes of various reactors, flowrates within the network, and concentrations of various components within those streams. The performance measure $g(\boldsymbol{x})$ is a twice continuously differentiable function. In many cases, this measure is only a function of the component concentrations at the exit of the reactor network, $\boldsymbol{c}^h$.

2.2.3. *Splitters*

Splitters are located at the network feed, and the outlets of each of the reactors. These constraints represent the overall mass balances around each unit. Due to the constant density assumptions, these balances can be written in terms of volumetric

flowrates.

$$F^{ac} + F^{ad} + F^{ah} - F^{a} = 0 \tag{2}$$

$$F^{ch} + F^{ec} + F^{cd} - F^{c} = 0 \tag{3}$$

$$F^{dh} + F^{dc} - F^{d} = 0 \tag{4}$$

2.2.4. *Mixers*

Mixers are located at the entrance to each of the reactors and at the outlet of the network. These constraints represent both the overall and component mass balances around each unit. Due to the constant density assumption, the overall balances can be written in terms of volumetric flowrates. The component balances are written in terms of moles.

$$F^{c} - F^{ec} - F^{dc} - F^{ac} = 0 \tag{5}$$

$$c_i^{c}\ F^{c} - c_i^{e}\ F^{ec} - c_i^{g}\ F^{dc} - c_i^{a}\ F^{ac} = 0\ \forall\, i\ \in\ I \tag{6}$$

$$F^{d} - F^{cd} - F^{ad} = 0 \tag{7}$$

$$c_i^{d}\ F^{d} - c_i^{e}\ F^{cd} - c_i^{a}\ F^{ad} = 0\ \forall\, i\ \in\ I \tag{8}$$

$$F^{a} - F^{ch} - F^{dh} - F^{ah} = 0 \tag{9}$$

$$c_i^{h}\ F^{a} - c_i^{e}\ F^{ch} - c_i^{g}\ F^{dh} - c_i^{a}\ F^{ah} = 0\ \forall\, i\ \in\ I \tag{10}$$

where I is the set of all components.

2.2.5. *CSTR*

The reaction rates, r_j, are known and are functions of the molar concentrations within the reactor:

$$r_j^{m} - f_j(\boldsymbol{c}^{g}) = 0\ \forall\, j\ \in\ J \tag{11}$$

where J is the set of all reactions, and the function $f_j(\boldsymbol{c}^{g})$ is twice continuously differentiable with respect to $\boldsymbol{c}^{g}$. The component balance, in terms of moles, around the reactor can be written as:

$$c_i^{g} - c_i^{d} - \tau^{m} \sum_{j \in J} \nu_{ij}\, r_j^{m} = 0\ \forall\, i\ \in\ I \tag{12}$$

where τ^{m} is the residence time of the reactor, and ν_{ij} is the stoichiometric coefficient of component i in reaction j.

2.2.6. *PFR*

The PFR is modeled under the assumption of perfectly plug flow. The reaction rates, $r_j^t(\bar{v})$, are functions of the molar concentration in the reactor, $\boldsymbol{c}^t(\bar{v})$.

$$r_j^t(\bar{v}) = f_j\left(\boldsymbol{c}^t(\bar{v})\right) \ \forall\, j \in J. \tag{13}$$

Both the concentration, and reaction rates are functions of $\bar{v}$, the scaled position along the length of the reactor. The component balance along the reactor, in terms of moles, is written in dynamic form:

$$\frac{\mathrm{d}c_i^t}{\mathrm{d}\bar{v}} = \tau^t \sum_{j\in J} \nu_{ij}\, r_j^t(\bar{v}) \ \forall\, i \in I \ \forall\, \bar{v} \in [0,1] \tag{14}$$

where τ^t is the residence time of the reactor. The initial conditions on the dynamic system are set by the inlet concentrations to the PFR:

$$c_i^t|_{\bar{v}=0} = c_i^c \ \forall\, i \in I. \tag{15}$$

In order to link the reactor back to the rest of the network, the following point constraints are imposed at the end of the PFR:

$$c_i^t|_{\bar{v}=1} = c_i^e \ \forall\, i \in I \tag{16}$$

2.2.7. *Formulation Notes*

There are some interesting properties of this formulation:

- Due to the dynamic modeling of the PFR, the formulation is truly a variable end time optimal control problem (OCP) with variable initial conditions. The OCP has been reformulated into a fixed end time problem by setting the independent variable in the integration ($\bar{v}$) to be 0 to 1 and scaling the dynamic equations by the residence time of the reactor τ^t. Equation (14) already reflects this reformulation.
- The formulation is highly nonconvex in nature. This is due to the bilinear nature in the mass balances around the mixers and the CSTR, the unknown nature of the expressions for the reaction rates in the CSTR (Equation 11), and the dynamic nature of the PFR. Therefore, this formulation may contain multiple local minima.

2.3. ADDITIONAL FORMULATION ELEMENTS

The formulation developed in Section 2.2 is sufficient to solve the network from a local point of view. To address the global optimization issue various additional

constraints and reformulations are introduced. Many of the additional constraints are redundant in nature, and were shown to be advantageous for the global solution based on the work of Quesada and Grossmann (1995) for mass networks. The advantages of these constraints in the reactor network formulation will be illustrated in Section 4.

2.3.1. *Splitter Component Balances*

It is possible to write balances around each of the splitters on a component basis. These constraints are satisfied by the overall mass balance around each splitter.

$$c_i^e \; F^{ch} + c_i^e \; F^{ec} + c_i^e \; F^{cd} - c_i^e \; F^c = 0 \; \forall \, i \; \in \; I \tag{17}$$

$$c_i^g \; F^{dh} + c_i^g \; F^{dc} - c_i^g \; F^d = 0 \; \forall \, i \; \in \; I \tag{18}$$

2.3.2. *Reactor Material Balances*

Material balances can be written around each of the reactors based on the stoichiometry of the given reaction mechanism. The material balances are satisfied by the reactor component balances. In general, the material balances take the form,

$$\sum_{i \in I} \gamma_i \left(c_i^g - c_i^d \right) = 0 \tag{19}$$

$$\sum_{i \in I} \gamma_i \left(c_i^e - c_i^c \right) = 0 \tag{20}$$

$$\sum_{i \in I} \gamma_i \left(F^d \, c_i^g - F^d \, c_i^d \right) = 0 \tag{21}$$

$$\sum_{i \in I} \gamma_i \left(F^c \, c_i^e - F^c \, c_i^c \right) = 0 \tag{22}$$

where γ_i are constants derived from the reaction stoichiometry. Notice that the first two of these constraints are linear in nature.

2.3.3. *Recycle Plug Flow Reactor*

In the superstructure given in Figure 1, the recycle around the PFR is explicitly included as a separate stream. As an alternative to that formulation, the PFR is replaced with the recycle reactor shown in Figure 2 and the stream F^{ec} is removed.

In the figure, R is the recycle ratio, defined as F^{ec}/F^c in the original superstructure. The following relationship for c_i^i, the inlet concentration to the plug flow section of the reactor, holds:

$$c_i^i = \frac{c_i^c + R \, c_i^e}{R + 1} \; \forall \, i \; \in \; I \tag{23}$$

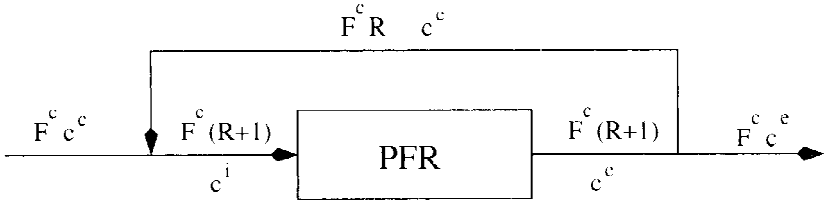


Figure 2. Recycle reactor.

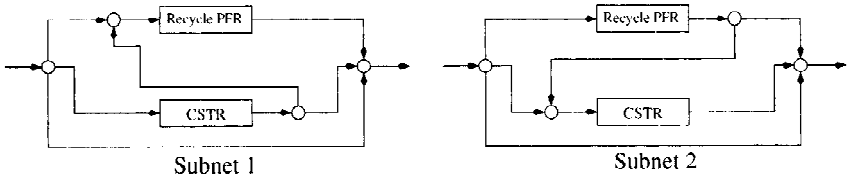


Figure 3. Sub-network structure.

Equation (23) is rewritten and included as a constraint in the formulation.

$$c_i^i + R\,c_i^i - c_i^c - R\,c_i^e = 0\;\forall\, i \in I \tag{24}$$

Due to the removal of the stream F^{ec} some constraints will need to be rewritten. The mass balances for the splitter after the recycle reactor become:

$$F^{ch} + +F^{cd} - F^c = 0 \tag{25}$$

$$c_i^e\ F^{ch} + c_i^e\ F^{cd} - c_i^e\ F^c = 0\;\forall\, i \in I \tag{26}$$

The mass balances for the mixer before the recycle reactor become:

$$F^c - F^{dc} - F^{ac} = 0 \tag{27}$$

$$c_i^c\ F^c - c_i^d\ F^{dc} - c_i^a\ F^{ac} = 0\;\forall\, i \in I \tag{28}$$

The initial conditions for the dynamic system for the plug flow section of the reactor are now written as:

$$c_i^t|_{\bar{v}=0} = c_i^i\;\forall\, i \in I \tag{29}$$

2.3.4. *Subnetwork Formulation*

In reality, at most one of crossing streams (F^{cd} and F^{dc}) would exist. Therefore, the network can be split into two different sub-networks. In each sub-network, only one of the crossing streams is included. The solution of these two smaller and simpler formulations takes much less time than the solution of the full superstructure. The two subnetworks are illustrated in Figure 3. Notice that by splitting the network, one splitter and one mixer are removed from each problem.

For each of the subnetworks, the formulation is changed significantly. Below is a concise set of the constraints for each of the two subnetworks. They include the additional formulation elements provided above (the redundant constraints, and recycle reactor reformulation).

subnetwork 1

Mixers

$$F^c - F^{dc} - F^{ac} = 0 \tag{30}$$

$$c_i^c\ F^c - c_i^g\ F^{dc} - c_i^a\ F^{ac} = 0\ \forall\, i\ \in\ I \tag{31}$$

$$F^a - F^c - F^{dh} - F^{ah} = 0 \tag{32}$$

$$c_i^h\ F^a - c_i^e\ F^c - c_i^g\ F^{dh} - c_i^a\ F^{ah} = 0\ \forall\, i\ \in\ I \tag{33}$$

Splitters

$$F^{ac} + F^d + F^{ah} - F^a = 0 \tag{34}$$

$$F^{dh} + F^{dc} - F^d = 0 \tag{35}$$

$$c_i^g\ F^{dh} + c_i^g\ F^{dc} - c_i^g\ F^d = 0\ \forall\, i\ \in\ I \tag{36}$$

CSTR

$$r_j^m - f(\boldsymbol{c}^g) = 0\ \forall\, j\ \in\ J \tag{37}$$

$$c_i^g - c_i^a - \tau^m \sum_{j\in J} \nu_{ij}\, r_j^m = 0\ \forall\, i\ \in\ I \tag{38}$$

$$\sum_{i\in I} \gamma_i \left(c_i^g - c_i^a\right) = 0 \tag{39}$$

Recycle PFR

$$r_j^t(\bar{v}) - f\left(\boldsymbol{c}^t(\bar{v})\right) = 0\ \forall\, j\ \in\ J \tag{40}$$

$$\frac{\mathrm{d}c_i^t}{\mathrm{d}\bar{v}} - \tau^t \sum_{j\in J} \nu_{ij}\, r_j^t(\bar{v}) = 0\ \forall\, i\ \in\ I\ \forall\, \bar{v}\ \in\ [0,1] \tag{41}$$

$$c_i^t|_{\bar{v}=0} - c_i^i = 0\ \forall\, i\ \in\ I \tag{42}$$

$$c_i^t|_{\bar{v}=1} - c_i^e = 0\ \forall\, i\ \in\ I \tag{43}$$

$$c_i^i + R\, c_i^i - c_i^c - R\, c_i^e = 0\ \forall\, i\ \in\ I \tag{44}$$

$$\sum_{i\in I} \gamma_i \left(c_i^e - c_i^c\right) = 0 \tag{45}$$

subnetwork 2

Mixers

$$F^d - F^{cd} - F^{ad} = 0 \tag{46}$$
$$c_i^d\ F^d - c_i^e\ F^{cd} - c_i^a\ F^{ad} = 0\ \forall\, i\ \in\ I \tag{47}$$
$$F^a - F^d - F^{ch} - F^{ah} = 0 \tag{48}$$
$$c_i^h\ F^a - c_i^e\ F^{ch} - c_i^g\ F^d - c_i^a\ F^{ah} = 0\ \forall\, i\ \in\ I \tag{49}$$

Splitters

$$F^{ad} + F^c + F^{ah} - F^a = 0 \tag{50}$$
$$F^{ch} + F^{cd} - F^c = 0 \tag{51}$$
$$c_i^e\ F^{ch} + c_i^e\ F^{cd} - c_i^e\ F^c = 0\ \forall\, i\ \in\ I \tag{52}$$

CSTR

$$r_j^m - f(\boldsymbol{c}^g) = 0\ \forall\, j\ \in\ J \tag{53}$$
$$c_i^g - c_i^d - \tau^m \sum_{j\in J} \nu_{ij}\, r_j^m = 0\ \forall\, i\ \in\ I \tag{54}$$
$$\sum_{i\in I} \gamma_i \left(c_i^g - c_i^d\right) = 0 \tag{55}$$

Recycle PFR

$$r_j^t(\bar{v}) - f\left(\boldsymbol{c}^t(\bar{v})\right) = 0\ \forall\, j\ \in\ J \tag{56}$$
$$\frac{dc_i^t}{d\bar{v}} - \tau^t \sum_{j\in J} \nu_{ij}\, r_j^t(\bar{v}) = 0\ \forall\, i\ \in\ I\ \forall\, \bar{v}\ \in\ [0,1] \tag{57}$$
$$c_i^t|_{\bar{v}=0} - c_i^i = 0\ \forall\, i\ \in\ I \tag{58}$$
$$c_i^t|_{\bar{v}=1} - c_i^e = 0\ \forall\, i\ \in\ I \tag{59}$$
$$c_i^i + R\, c_i^i - c_i^a - R\, c_i^e = 0\ \forall\, i\ \in\ I \tag{60}$$
$$\sum_{i\in I} \gamma_i \left(c_i^e - c_i^a\right) = 0 \tag{61}$$

By splitting the superstructure into these two subnetworks, the resulting formulations are simplified by more than just the reduction in the numbers of mixers and splitters. In each of the subnetwork formulations, one of the reactors is also simplified.

- In Subnetwork 1, the inlet of the CSTR is now a constant composition, $\boldsymbol{c}^a$, which is the feed composition to the network.

- In Subnetwork 2, the inlet of the Recycle PFR is a constant composition, $\boldsymbol{c}^a$. Therefore the initial conditions on the dynamic system, c_i^i, are now only functions of the recycle ratio, R. As will be shown in the next section, this simplification makes the global solution of the problem much easier.

2.3.5. *Reduced Component Formulation*

Only a subset of the components needs to be accounted for. A subset of components, I', is defined to contain those components meeting the following conditions:

1. Appearing in the performance measure, $g(\boldsymbol{x})$
2. Appearing in the reaction rate expressions, $f_j(\boldsymbol{c})$ in equations (11) and (13).
3. Reactants which do not appear in the rate expressions.

All the other component concentrations are not necessary for the solution of the optimization problem. If their concentrations are needed, they can be determined by a simple set of function evaluations using the solution to the optimization problem. The formulation is changed by rewriting all the constraints defined over the set I, over the set I'. By reducing the number of components used in the formulation, the number of constraints is reduced by half in some cases. This will reduce the time it takes to determine a local solution to the formulation.

There is one other result of this reformulation. Since all of the components are no longer included, the reactor material balance Eqs. (19) – (22) need to be rewritten. All of the reactants are included, but not all the products. Therefore, the total amount of reactants consumed is known, which must be greater than or equal to the amount of products included in I' produced. The balances are rewritten as inequality constraints:

$$\sum_{i \in I'} \gamma_i \left(c_i^g - c_i^d \right) \leqslant 0 \tag{62}$$

$$\sum_{i \in I'} \gamma_i \left(c_i^e - c_i^c \right) \leqslant 0 \tag{63}$$

$$\sum_{i \in I'} \gamma_i \left(F^d \, c_i^g - F^d \, c_i^d \right) \leqslant 0 \tag{64}$$

$$\sum_{i \in I'} \gamma_i \left(F^c \, c_i^e - F^c \, c_i^c \right) \leqslant 0 \tag{65}$$

2.4. FORMULATION CLASSES

In the solution of the example problems, four different classes of formulations were investigated. For clarity, the elements used in each of these formulations are described below:

1. **Standard**. This is the standard formulation containing the constraints defined in Section 2.2 only. As previously mentioned, this formulation did not perform well in the global optimization approach.
2. **Standard w/ additional constraints**. This formulation includes all the elements of the standard formulation, plus the constraints defined in Sections 2.3.1 and 2.3.2.
3. **Recycle PFR**. This formulation includes the standard w/ additional constraints formulation, plus the recycle PFR reformulation defined in Section 2.3.3. This formulation is solved as a full network or using the sub-network structure defined in Section 2.3.4.
4. **Reduced component**. This formulation includes all the elements of the recycle PFR formulation, but reduces the number of components used as described in Section 2.3.5. This formulation is solved as a full network or using the sub-network structure defined in Section 2.3.4.

3. Global Optimization Approach

Within this section, a novel global optimization approach for the solution of the reactor network synthesis problem will be presented. The approach is based on the αBB method presented by Adjiman et al. (1998a, b), with extensions for optimal control formulations given by Esposito and Floudas (2000a). First, the basic concepts of the αBB will be given, then various extensions to tailor the approach to this particular problem will be discussed.

3.1. BASIC CONCEPTS OF THE αBB

The αBB global optimization method (Androulakis et al. 1995; Adjiman and Floudas, 1996; Adjiman et al. 1996, 1998a, b) guarantees convergence to an ϵ-global minimum for general twice continuously differentiable constrained and unconstrained NLPs. The applicability of the technique to differential-algebraic systems has recently been shown by Esposito and Floudas (2000a, b). The approach generates a non-decreasing sequence of lower bounds and a non-increasing sequence of updated upper bounds on the global solution. Finite ϵ-convergence to the global minimum is achieved through the successive subdivision of the region at each level in the branch-and-bound tree. The sequence of upper bounds on the global solution is obtained by solving, to local optimality, the full nonconvex problem from different starting points. Lower bounds are generated by solving a convex relaxation which underestimates the original problem. The overall steps in the approach are outlined below.

1. Parse the problem, generating necessary information such as analytical first and second order derivatives.
2. Generate a valid convex relaxation which underestimates the original formulation.

3. Update the bounds on some or all of the algebraic variables by solving a sequence of feasibility problems.
4. Solve the relaxed problem to local optimality to generate an initial lower bound on the global solution. Save this solution and variable bounds to be explored later.
5. Solve the original nonconvex problem to local optimality to generate an upper bound on the global solution.
6. Select the next saved region to be explored as the one with the lowest lower bound.
7. Check for global convergence. If the lower bound and upper bound are within a specified tolerance, stop and declare the upper bounding solution the global solution to the problem. Otherwise, continue.
8. Update the bounds on some or all of the algebraic variables by solving a sequence of feasibility problems.
9. Branch the region into two by bisecting on a selected variable.
10. In each new region, solve the relaxed problem to local optimality. If the solution is greater than the current upper bound, or no feasible solution is found, then reject the region since it cannot contain the global solution. If the solution is less than the current upper bound, save the region for later exploration.
11. In each new region, solve the original nonconvex problem to local optimality using the solution of the relaxed problem as a starting point. If the solution is less than the current upper bound, update the upper bound.
12. Go to step 6

This framework makes up the generic αBB approach. There are many parts which allow for the customization of the method to a given class of problems. These include: (1) the formulation of the problem, (2) the generation of the convex relaxation, (3) the selection of the branching variable, and (4) the method used to update the variable bounds. The problem formulation was previously discussed in Section 2. The last three elements will be discussed in the following sections.

3.2. CONVEX RELAXATION

A key aspect of this branch-and-bound approach is the ability to generate a valid convex relaxation which underestimates the original formulation. This is accomplished by replacing each of the nonconvex terms in the algebraic constraints and the objective function by a convex underestimator. Looking at the formulation of the problem, most of the nonconvexities take the form of bilinearities. They are underestimated by a method given by Al-Khayyal and Falk (1983), Al-Khayyal (1990), and McCormick (1976). Each bilinear term, xy, is replaced by an auxiliary

variable, w, and the following four linear inequalities are added.

$$\begin{array}{rl} x^l y + y^l x - x^l y^l - w & \leqslant 0 \\ x^u y + y^u x - x^u y^u - w & \leqslant 0 \\ -x^u y - y^l x + x^u y^l + w & \leqslant 0 \\ -x^l y - y^u x + x^l y^u + w & \leqslant 0 \end{array} \tag{66}$$

It should be noted that any constraint that contains only linear and bilinear terms will become linear in the relaxed formulation. Additional nonconvexities can occur within the rate expressions for the reaction mechanism, $\mathbf{f}(\boldsymbol{c}^s)$ given in (11). The types of terms which occur in this expression are not known. For a univariate concave term $u(x)$, for example $-x^2$, the tightest possible underestimator, $\mathcal{L}(x)$, is a linearization from x^L to x^U.

$$\mathcal{L}(x) = u(x^L) + \frac{u(x^U) - u(x^L)}{x^U - x^L}(x - x^L) \tag{67}$$

Terms which do not fall under one of the two types presented above, are relaxed using an α underestimator developed by Maranas and Floudas (1994). For a nonconvex term in several variables, $NC(\boldsymbol{x})$, the underestimator, $\mathcal{L}(\boldsymbol{x})$, would be

$$\mathcal{L}(\boldsymbol{x}) = NC(\boldsymbol{x}) + \alpha \sum_{i \in \mathcal{X}} (x_i^U - x_i)\,(x_i^L - x_i) \tag{68}$$

where $\mathcal{X}$ is the set of $\boldsymbol{x}$ variables participating in the term $NC(\boldsymbol{x})$. The value of α needs to be large enough to generate a convex function, but not too large as to overly underestimate the function. It is shown that,

$$\begin{aligned} \alpha \geqslant & \max\left\{0, -\frac{1}{2}\min_k \lambda_k(\boldsymbol{x})\right\} \\ & \text{s.t.} \quad \boldsymbol{x}^L \leqslant \boldsymbol{x} \leqslant \boldsymbol{x}^U \end{aligned} \tag{69}$$

where $\lambda_k(\boldsymbol{x})$ are the eigenvalues of the Hessian matrix of $NC(\boldsymbol{x})$. It is preferable to derive an analytical expression for the value of α using (69) as an equality. This will provide the tightest possible convex underestimation of $NC(\boldsymbol{x})$. When this is not possible, interval mathematical methods are available for the generation of valid α values as shown by Adjiman and Floudas (1996) and Adjiman et al. (1998b).

The dynamic part of the formulation (the constraints which describe the PFR or recycle PFR), is underestimated using a method presented by Esposito and Floudas (2000a). The dynamic system is considered as a simple input/output map:

$$\begin{matrix} \boldsymbol{c}^i \\ \tau^t \end{matrix} \longrightarrow \boxed{\begin{matrix} \text{DAEsystem} \\ \text{Eqs. (13) – (16)} \end{matrix}} \longrightarrow \boldsymbol{c}^e \tag{70}$$

The input to this map is the initial conditions on the dynamics (the inlet concentration to the PFR) and the residence time of the reactor. The output of the map is the

outlet concentrations of the various components. This map can be also written as a set of functions,

$$c_i^e = \mathcal{F}_i(\boldsymbol{c}^i, \tau^t) \; \forall \, i \; \in \; I. \tag{71}$$

Given the conditions placed on the function $f(\boldsymbol{c})$ used to describe the kinetics of the reaction mechanism, the function defined by (71) is twice continuously differentiable with respect to $\boldsymbol{c}^i$ and τ^t. The continuity and differentiability with respect to the parameter, τ^t, has been illustrated by Esposito and Floudas (2000a). The continuity and differentiability with respect to the initial conditions, $\boldsymbol{c}^i$, is illustrated below.

Pontryagin (1962) provides a series of theorems concerning the continuity and differentiability of the solutions of the following type of system:

$$\dot{z} = \mathbf{g}(z, \boldsymbol{v}, t), \tag{72}$$

The system of DAEs given for the PFR in Section 2.2.6 can be converted into the form given in (72) by substituting the reaction rates, $r_j^t(\bar{v})$, into the dynamic equations given by (14). It is also assumed that the right hand side of (72), $\mathbf{g}(z, \boldsymbol{v}, t)$, and the partial derivatives,

$$\frac{\partial}{\partial z}\mathbf{g}(z, \boldsymbol{v}, t) \tag{73}$$

are defined and are continuous in some domain $\boldsymbol{\Gamma}$ of the space of variables t, z, and $\boldsymbol{v}$. These conditions will hold because of the nature of the PFR equations and the twice continuous differentiability condition placed on the reaction rate expressions given in (13). This leads to the following:

THEOREM 1 (Pontryagin, 1962, p. 179). *If (t_0, z_0) is an arbitrary point of the domain $\boldsymbol{\Gamma}$, there exist positive numbers r' and σ' such that the solution*

$$z = \boldsymbol{\psi}(t; \boldsymbol{v}, \boldsymbol{\eta})$$

of (72) which satisfies the initial condition:

$$\boldsymbol{\psi}(t_0; \boldsymbol{v}, \boldsymbol{\eta}) = \boldsymbol{\eta}$$

is defined and continuous in the variables, t and $\boldsymbol{\eta}$ for

$$|t - t_0| \leqslant r' |\boldsymbol{\eta} - z_0| \leqslant \sigma'$$

and has continuous partial derivatives with respect to $\boldsymbol{\eta}$.

COROLLARY 1 (Pontryagin, 1962, p. 179). *If all the partial derivatives of $\mathbf{g}(z, \boldsymbol{v}, t)$ with respect to the variables z and $\boldsymbol{v}$ up to the m^{th} order inclusive exist and are continuous, then the functions $\boldsymbol{\psi}(t, \boldsymbol{v}, \boldsymbol{\eta})$ also have continuous partial derivatives with respect to $\boldsymbol{\eta}$, up to the m^{th} order inclusive.*

Therefore, the function $\mathcal{F}(\boldsymbol{c}^i, \tau^t)$ is also twice continuously differentiable with respect to the initial conditions, $\boldsymbol{c}^i$

$\mathcal{F}(\boldsymbol{c}^i, \tau^t)$, as well as it's first and second order derivatives, are able to be evaluated using an integration routine. These functions are underestimated using a similar approach to the α method shown in (68).

$$\begin{aligned}\mathcal{L}_i(\boldsymbol{c}^i, \tau^t) =& \mathcal{F}_i(\boldsymbol{c}^i, \tau^t) + \sum_{j \in \mathcal{I}} \beta^c_{i,j} \, (c^{i,U}_j - c^i_j) \, (c^{i,L}_j - c^i_j) \\ &+ \beta^\tau_i \, (\tau^{t,U}_i - \tau^t) \, (\tau^{t,L} - \tau^t) \end{aligned} \tag{74}$$

A simplification of this form involves the use of a single β parameter for all the variables involved in the expression:

$$\begin{aligned}\mathcal{L}_i(\boldsymbol{c}^i, \tau^t) =& \mathcal{F}_i(\boldsymbol{c}^i, \tau^t) + \beta_i \left\{ \sum_{j \in \mathcal{I}} (c^{i,U}_j - c^i_j) \, (c^{i,L}_j - c^i_j) \right. \\ &\left. + \, (\tau^{t,U}_i - \tau^t) \, (\tau^{t,L} - \tau^t) \right\} \end{aligned} \tag{75}$$

Esposito and Floudas (2000a, b) detailed methods of determining valid values for the β parameters in (74) and (75). These methods are summarized here. First, the Hessian matrix, $\mathcal{H}_i$, of the function $\mathcal{F}_i(\boldsymbol{c}^i, \tau^t)$, is defined as:

$$\mathcal{H}_i \equiv \frac{\partial^2 \mathcal{F}_i}{\partial \boldsymbol{v}^2} \tag{76}$$

where $\boldsymbol{v}$ is the vector $[\boldsymbol{c}^i, \tau^t]$. The elements of this matrix are not analytical functions of the variables. Instead, the matrix elements are determined at a given value of $[\boldsymbol{c}^i, \tau^t]$ through the integration of an augmented dynamic system. As a result of the implicit nature of the Hessian matrices, three different methods for determining β values have been developed:

1. **Constant**. Constant values are preselected for each of the β parameters. Either one β parameter for each term (Eq. (75)), or one β parameter for each variable in each term (Eq. (74)) can be used.
2. **Sampling**. For this approach, only one β per term can be used. Before the first iteration, a given number of points, p^{init}, in the variable space are selected and the Hessian matrices, $\mathcal{H}_{p,i}$ are evaluated. The minimum eigenvalue of each matrix, $\lambda^{\min}_{p,i}$, is determined and the β parameters are calculated by:

$$\begin{aligned}&\beta_i = -\frac{1}{2} \min_p \lambda^{\min}_{p,i} \\ &\beta_i \geqslant 0 \end{aligned} \tag{77}$$

   These β parameters are updated each time the bounds on any of the variables $\boldsymbol{c}^i$ or τ^t change. At least a given number of points, p^{every}, is used in each calculation.

3. **Sampling/Interval Analysis**. In this method, once again the Hessian matrices, $\mathcal{H}_{p,i}$ are evaluated at a given number of points. This time though, an interval Hessian matrix, $[\mathcal{H}]_i$, is created from the minimum and maximum of each element over the set of sampled points. Interval methods proposed by Adjiman and Floudas (1996) and Adjiman et al. (1998b) are then used to generate β values. Either a single β per term or a β per variable per term can be calculated.

3.3. SELECTION OF A BRANCHING VARIABLE

At each level in the tree, it is necessary to select a variable to branch the current region on. In a standard method, this is accomplished by selecting a variable based on its overall contribution to the quality of the convex relaxation.

$$j^* = \arg\max_{j \in J} \delta_j \tag{78}$$

where j^* is the index of the branching variable, and δ_j is the overall contribution calculated for a given variable j. The set J represents the list of variables which are considered for branching. This set may contain any number of variables in the problem. The overall contribution is made up of two components,

$$\delta_j = \delta_j^d + \delta_j^a \tag{79}$$

where δ_j^d is the measure calculated from the dynamic part of the formulation, and δ_j^a is from the algebraic part. These measures are determined by calculating a variable's contribution to the separation between each nonconvex term and it's underestimator at the solution to the relaxed problem in the current region. For the dynamic part of the formulation, the measure is calculated by

$$\delta_j^d = \sum_{i \in I} \beta_{i,j} \ \left(v_j^U - v_j^{\text{sol}}\right)\left(v_j^L - v_j^{\text{sol}}\right) \tag{80}$$

where $\boldsymbol{v}$ again represents the vector $[\boldsymbol{c}^i, \tau^t]$, and $\boldsymbol{v}^{\text{sol}}$ is its value at the solution to the relaxed problem. For the algebraic part, this measure is calculated as:

$$\delta_j^a = \sum_{k \in K_j} f_k(\boldsymbol{x}^{\text{sol}}) - \mathcal{L}_{f_k}(\boldsymbol{x}^{\text{sol}}) \tag{81}$$

where K_j is the set of algebraic functions, $f_k(\boldsymbol{x})$, in which the given variable j participates and $\mathcal{L}_{f_k}(\boldsymbol{x})$ represents the underestimator of the function.

As an extension to this method, consider a case in which each of the variables in the branching set can be given a weight, $\mathcal{W}_j$, which represents its importance in the problem formulation. Equation (78) is then rewritten as,

$$j^* = \arg\max_{j \in J} \delta_j \ \mathcal{W}_j. \tag{82}$$

These weights could simply be set at the beginning of the algorithm. This would require a substantial amount of prior knowledge of how the algorithm is effected by branching on a certain variable. In order to overcome this difficulty, an *adaptive* method, which calculates the variable weights depending on how the algorithm is performing, was developed.

At each iteration of the global optimization approach two child regions are generated from a given parent by branching on a selected variable. The true objective in selecting a branching variable is to choose the variable which will result in the greatest increase in the lower bound in each of the child regions. The selection method shown above, attempts to do this, but does not always select the best variable. In this adaptive approach, the weights on the variables will be changed depending on their branching performance, i.e., the increase in the lower bound they generate. The approach calculates the relative change in the lower bound from the parent to each child. If the change is greater than a *nominal* value, the weighting is increased, and if it is less, the weighting is decreased. Also, some minimum and maximum weighting must be enforced. The approach, as implemented, is detailed below:

1. Set all $\mathcal{W}_i$ equal to 1 initially, chose an adaption parameter, A. This parameter should be between 0 and 1. It determines the nominal increase in the lower bound required, and the size of the weighting change on a given variable.
2. At each iteration, and for each region, calculate the relative change, C^{lb}, in the lower bound between the current region and its parent. Also calculate the relative difference between the upper bound and the current lower bound, ϵ^{rel}. The index of the branched variable is i.
3. Three different cases are possible:

   If the relaxed problem is infeasible, or the region is rejected, $\mathcal{W}_i = \mathcal{W}_i + A$.
   Else if $C^{lb} < A \times \epsilon^{\text{rel}}$ then

   if $\mathcal{W}_i \leqslant 1$ then $\mathcal{W}_i = \mathcal{W}_i - A$
   if $\mathcal{W}_i > 1$ then $\mathcal{W}_i = 1$

   Else if $C^{lb} > A \times \epsilon^{\text{rel}}$ then $\mathcal{W}_i = \mathcal{W}_i + A$

4. Enforce bounds of [0.25, 3] on $\mathcal{W}_i$

The advantage of this method will be illustrated in Section 4.

3.4. VARIABLE BOUNDS UPDATING

At each iteration of the approach, the bounds on all or a subset of variables can be updated by solving a sequence of feasibility problems. These problems are

formulated as such,

$$x_j^{L,*}/x_j^{U,*} = \begin{cases} \min\limits_{\boldsymbol{x}} \ / \ \max\limits_{\boldsymbol{x}} & x \\ \text{s.t.} & \mathcal{L}(\boldsymbol{x}) \leqslant 0 \\ & \mathcal{L}_g(\boldsymbol{x}) \leqslant UB \\ & \boldsymbol{x}^L \leqslant \boldsymbol{x} \leqslant \boldsymbol{x}^U \end{cases} \tag{83}$$

where $\boldsymbol{x}$ is the set of all algebraic variables, and $j \in J$ where J is the set of variables selected for updating. $\mathcal{L}(\boldsymbol{x})$ represents the relaxed constraint set, $\mathcal{L}_g(\boldsymbol{x})$ is the relaxed objective function, and UB is the current upper bound on the global solution. Formulation (83) includes all of the constraints in the convex relaxation, including the dynamic equations. Considering a simple four component system, there are at least 25 variables that appear in nonconvex constraints. Updating the bounds on all of these variables at each iteration would require the solution of (83) over 50 times. The computational effort required can be substantial. Therefore a reduced formulation including only the linear constraints is generated:

$$x_j^{L,*}/x_j^{U,*} = \begin{cases} \min\limits_{\boldsymbol{x}} \ / \ \max\limits_{\boldsymbol{x}} & x \\ \text{s.t.} & \mathcal{L}^{\text{lin}}(\boldsymbol{x}) \leqslant 0 \\ & \mathcal{L}_g^{\text{lin}}(\boldsymbol{x}) \leqslant UB \\ & \boldsymbol{x}^L \leqslant \boldsymbol{x} \leqslant \boldsymbol{x}^U \end{cases} \tag{84}$$

where $\mathcal{L}^{\text{lin}}(\boldsymbol{x})$ is the set of linear relaxed constraints, and $\mathcal{L}_g^{\text{lin}}(\boldsymbol{x})$ is the relaxed objective function if it is linear. Due to the nature of the formulation, and the type of underestimators used, $\mathcal{L}^{lin}(\boldsymbol{x})$ includes almost all of $\mathcal{L}(\boldsymbol{x})$. As noted in Section 3.2, any constraint including only bilinear and linear terms will be linear in the relaxation. Therefore, all stream balances, reactor material balances, CSTR mass balances, part of the recycle PFR mass balance, and possibly part of the CSTR reaction expressions depending on the type of kinetics, are included in the constraint set. The formulation is not as tight, and therefore does not result in as large a reduction in the variable ranges as if the full relaxed constraint set was used. However, the solution of a linear problem is substantially more computationally efficient than a dynamic problem. The advantages of using this reduced formulation will be illustrated in Section 4.

Another level of simplification can be introduced by identifying variables whose bounds can directly be calculated from bounds on other variables. The variables for which explicit expressions can be written are $\boldsymbol{c}^i$ and $\boldsymbol{r}^m$. These expressions take the form:

$$c_i^{i,U} = \frac{1}{1+R^L}\, c_i^{c,U} + \frac{R^U}{1+R^U}\, c_i^{e,U}\, r_j^{m,U} = f_j^U(\boldsymbol{c}^g)$$

$$c_i^{i,L} = \frac{1}{1+R^U}\, c_i^{c,L} + \frac{R^L}{1+R^L}\, c_i^{e,L}\, r_j^{m,L} = f_j^L(\boldsymbol{c}^g)$$

where $f_j^U(\boldsymbol{c}^g)$ and $f_j^L(\boldsymbol{c}^g)$ are the upper and lower bounds on the function $f_j(\boldsymbol{c}^g)$. These functional bounds can be determined analytically, or using interval analysis.

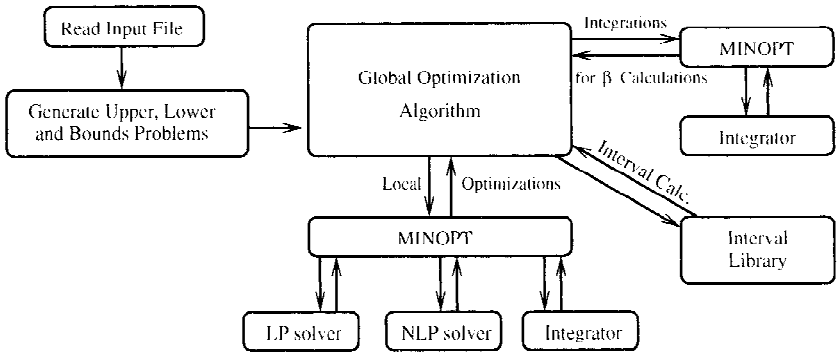


Figure 4. Program flow.

3.5. IMPLEMENTATION

This algorithmic procedure has been implemented in an extensive C program with an intuitive front end parser. All of the various formulations (original, relaxed, linear bounds updating) are generated automatically. A link to the MINOPT optimization program (Schweiger and Floudas, 1998) is used to perform the local optimizations and integrations. MINOPT itself has links to various local solvers. For these problems, NPSOL (Gill et al., 1986) was used as the local nonlinear optimization routine, CPLEX (CPLEX, 1997) is used to solve the linear bounds updating problems, and DAESSA (Schweiger and Floudas, 1998) performs all the necessary dynamic integrations. Also a link to the interval math library, PROFIL/BIAS, is used in the automatic determination of β values. Figure 4 illustrates the flow of information within the implementation.

4. Computational Studies

Five different isothermal reactor network problems will be presented to illustrate the performance of the proposed approach. All computational results were obtained on a Pentium III/600 running Linux. A common set of conditions for the algorithm for each example include the following:

1. *Branching*: The variables included in the branching set are the flowrates and component concentrations which appear in bilinear terms within the mass balances, as well as the residence time of the reactors and the recycle ratio ($\boldsymbol{c}^g$, $\boldsymbol{c}^e$, $\boldsymbol{c}^d$, $\boldsymbol{c}^c$, F^c, F^{cd}, F^{ch}, F^d, F^{dc}, F^{dh}, τ^m, τ^t, R).
2. *Bounds Updating*: The same variables included in the branching set will also be included in the updating set. The variables $\boldsymbol{c}^i$ and $\boldsymbol{r}^m$, which are not in the updated set, will have their bounds calculated at each iteration as a function of the bounds on other variables (as shown in Section 3.4).
3. *Variable Bounds*: The bounds on the flowrates and the recycle ratio, R, are set to [0, 1]. The bounds on the component concentrations are set depending on the

Table 2. Local solutions for Van De Vusse, Case 1 example.

Obj. value	Configuration	Frequency (%)
0.43708	PFR (0.2533 s)	67.2
0.43309	CSTR (0.05 s) + PFR (0.2138 s)	4.6
0.33133	CSTR (0.3208 s)	5.1
Failed	–	22.6

inlet concentration and the reaction stoichiometry. Bounds on residence times are also different for each problem studied. It should be noted that the bounds on the residence times are set strictly positive. This forces the nonexistence of a reactor to be indicated when there is no flow to or from it. This removes redundancy in which a zero size reactor simply would act as a transfer line.

4.1. VAN DE VUSSE REACTION, CASE 1

The Van de Vusse reaction mechanism has four species and three different reactions:

$$A \xrightarrow{k_1} B \xrightarrow{k_2} C$$

$$2A \xrightarrow{k_3} D$$

The rate expressions for this mechanism take the form:

$$f_1 = k_1\, c_A$$
$$f_2 = k_2\, c_B$$
$$f_3 = k_3\, c_A^2$$

The inlet stream to the network is pure A, with a concentration of 0.58 mol/l. The rate constant vector used is:

$$k = [10\mathrm{s}^{-1}, 1\mathrm{s}^{-1}, 0.5\mathrm{l/mol\,s}].$$

The objective is to maximize the outlet concentration of component B. The bounds on the concentrations are set to: $c_A, c_B, c_C \in [0, 0.58]$; $c_D \in [0, 0.29]$. The bounds on the residence time of the reactors are set at: $\tau^t, \tau^m \in [0.05\mathrm{s}, 1\mathrm{s}]$. Using the recycle PFR formulation, 4 different local solutions were identified, which are shown in Table 2. The frequency of the solution is the percentage of randomly chosen starting points that resulted in the given objective value. Notice that the local solver failed almost 25 % of the time.

Table 3. Global Solution times for the Van De Vusse, Case 1 example using different formulation classes.

Formulation	Sub-network	Iterations	CPU (min)
Standard	–	5000*	697.35
Additional constraints	–	449	102.40
Recycle PFR	Full	347	75.70
	Subnetwork 1	129	20.83
	Subnetwork 2	41	5.28
Reduced component	Full	1181	63.99
	Subnetwork 1	285	9.67
	Subnetwork 2	139	4.58
Non-Recycle PFR	Full	98	24.87
	Subnetwork 1	75	10.86
	Subnetwork 2	14	1.74
Reduced component	Full	183	8.93
(Nonrecycle PFR)	Subnetwork 1	87	2.32
	Subnetwork 2	24	0.75

* Only achieved a relative difference between the upper and lower bounds of 2.15 %, but the upper bound was the known global solution.

Using the standard branching method and full dynamic bounds updating problems, this example was solved to global optimality (1% relative convergence tolerance) using each of the different formulations presented in Section 2.4. A constant value of 0.1 was used for all the β values needed in the formulation. The results are shown in Table 3. Notice that the standard formulation does not lead to convergence in 5000 iterations. By simply adding the redundant constraints, convergence is achieved in under 2 CPU hours. If the Recycle PFR reformulation is then used, this time is reduced to 1.25 hours. By splitting the network and solving both subnetworks to global optimality, the time is cut to 25 min. Using only the principle components in the formulation, the solution time is finally reduced to 15 min. By using the various reformulations the solution time is reduced from over 11 h to 15 min.

Also included in Table 3 are results when the recycle around the PFR is removed. These results illustrate the complication a recycle stream adds to the formulation. For instance, using the full network formulation, with all four components, the solution time is reduced from 75 to 25 min. by simply removing the recycle stream. In the attainable region approach, it is argued that in a two-dimensional reactor network synthesis problem only CSTRs and PFRs (without recycle) are needed for an optimal (in concentration space) configuration (Hildebrandt et al., 1990). This example is two-dimensional, optimized in the concentration space, and

Table 4. Global solution times for Van De Vusse, Case 1 example using the sampled with interval analysis β calculation method. 1000 points were used initially, and at least 10 in each subsequent region.

Formulation	Sub-network	Iterations	CPU (min)
Recycle PFR	Full	532	168.60
	Subnetwork 1	190	44.88
	Subnetwork 2	75	12.83
Reduced Component	Full	2100	99.50
	Subnetwork 1	708	17.97
	Subnetwork 2	453	11.52

the global solution does not include a recycle around the PFR. Nevertheless, the recycle around the PFR is included in the superstructure for completeness. If the objective was based on cost measures, which include the size of the reactors, the validity of the attainable region theory is unknown, and the global solution may contain a recycle around the PFR.

The results presented to this point are obtained using constant values for the β parameters which may or may not be valid since no second order information is being used. A value which ensures convexity of the lower bounding problem is not always required to obtain convergence to the global minimum (Esposito and Floudas, 2000a). Nevertheless, to prove convergence to the global minimum, a valid convex underestimator is needed. Table 4 shows results for the final two formulations using the sampling with interval analysis β calculation method detailed in Section 3.2. To reasonably ensure that a convex problem is generated, 1000 points are used initially (p^{init}) and at least 10 in each subsequent region (p^{every}). The scaled Gerschgorin theorem was used to determine the minimum eigenvalue of the interval Hessian matrix (Adjiman et al., 1998b). The iteration count and the solution time both increase from the constant case. The increase in the iteration count is a result of an increase in the β values used which results in a 'looser' relaxation requiring more iterations to reach convergence. The increase in the computational time is a result of both the increase in the number of iterations, and the time required to generate the necessary second-order information. This extra computational effort is required to meet the theoretical conditions to prove convergence to the global minimum.

The next algorithmic measure to be explored is the use of linear bounds updating problems. The example was solved using the same conditions used to generate Table 4 with the exception of linear bounds updating problems. Table 5 shows the results using both the full and reduced component formulations. By using the linear bounds updating problems, the iterations required to converge to the global solution are greatly increased. This is simply the result of not determining the tightest best

Table 5. Global solution times for Van De Vusse, Case 1 example using the linear bounds updating problems.

Formulation	Sub-network	Iterations	CPU (min)
Recycle PFR	Full	2803	74.74
	Subnetwork 1	1367	27.60
	Subnetwork 2	232	6.00
Reduced Component	Full	8982	91.04
	Subnetwork 1	2566	11.71
	Subnetwork 2	969	4.49

possible bounds on each variable at each iteration. Nevertheless, the computational time is reduced since the linear bounds updating procedure only requires a fraction of the time of the dynamic procedure. There is one anomalous result presented in Table 5. In solving the reduced component formulation for the full network, the use of linear bounds problems does not result in a significant reduction in time. In this case, due to the large number of iterations (nearly 9000), a significant amount of time is spent performing second order sensitivity evaluations in the calculation of β values. In all other cases in both Tables 4 and 5, these evaluations require 1–2 CPU min, but in this case 28 CPU min were required. If we remove this element of the computational time, the solution would require 63 CPU min. This is a reduction of over 30 CPU min from the dynamic updating case.

Finally, the selection of the branching variable will be looked at. The problem is solved using the adaptive branching scheme presented in Section 3.3 (with a constant $A = 0.25$) along with the linear bounds updating problems. Table 6 shows the results using both the full and reduced component formulations. In most cases, the use of the adaptive branching selection scheme results in a small or no reduction in the number of iterations. However, in the case of the first subnetwork formulation, the reduction in iteration count is quite significant. In the reduced component formulation, a reduction of about 40% in both the iteration count and computational time is observed. Figure 5 shows the frequency with which certain variables are selected for branching. The differences between the standard scheme and the adaptive scheme are small, but significant.

This example clearly illustrates the advantages of the various algorithmic improvements. Using the standard algorithmic conditions with the reduced component formulation and subnetwork split, the global solution is obtained in a total of 29.49 CPU min. The application of the linear bounds updating problems and the adaptive branching scheme result in the computational effort being cut more than in half to 12.01 CPU min. A *tuned* set of algorithmic conditions can then be defined. These conditions include the use of the adaptive branching scheme with $A = 0.25$ and the linear bounds updating problems. Additionally, calculated β values are

Table 6. Global solution times for the Van De Vusse, Case 1 example using the adaptive branching scheme and linear bounds updating problems.

Formulation	Sub-network	Iterations	CPU (min)
Recycle PFR	Full	2835	71.15
	Subnetwork 1	1163	19.25
	Subnetwork 2	253	6.47
Reduced Component	Full	8727	90.53
	Subnetwork 1	1480	7.45
	Subnetwork 2	925	4.56

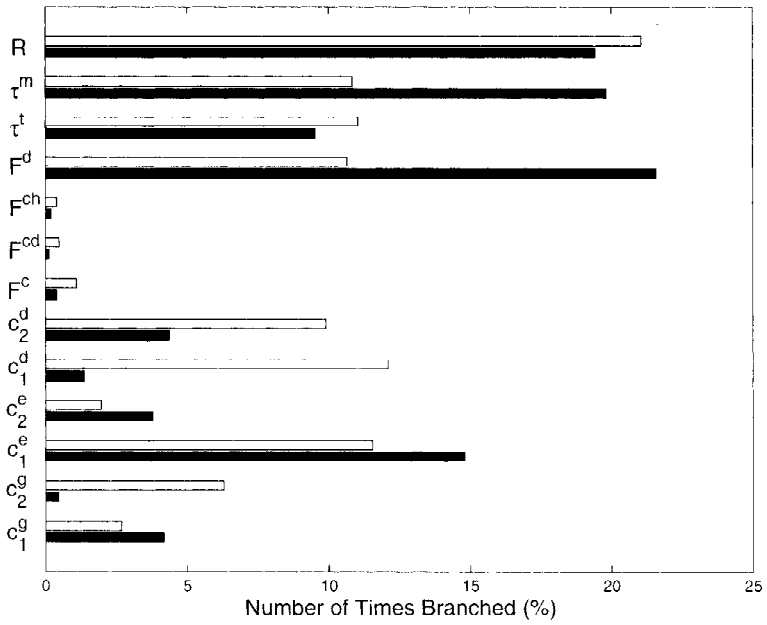


Figure 5. Comparison of the branching selection frequency for the Van De Vusse, Case 1 example using the standard (empty bars) and the adaptive (filled bars) branching schemes.

determined using the scaled Gerschgorin theorem with a sample size of 1000 points initially (p^{init}) and at least 10 in each subsequent region (p^{every}). These algorithmic parameters will be used to solve all subsequent examples.

It is also interesting to note that for each of the cases solved in this example, the upper bound determined at the root node of the tree was in fact the known global solution. This initial solution requires only a few seconds of CPU time to generate. The rest of the computational effort is required to prove the global optimality of

Table 7. Local solutions for Van De Vusse, Case 2 example.

Obj. value	Configuration	Frequency (%)
0.3682	CSTR (0.1135 s) + PFR (0.1699 s)	44.9
0.3586	PFR (0.2152 s ; R = 0.206)	29.0
0.3573	PFR (0.1588 s ; R = 0.458) + CSTR (0.05 s)	1.3
0.3061	CSTR (0.3560 s)	12.5
0	All Bypass	8.2
Failed	–	12.3

that solution. Therefore, the algorithm acts as a highly effective structured search requiring little effort to actually identify the global solution. The rest of the time is required to prove global optimality.

4.2. VAN DE VUSSE REACTION, CASE 2

This example uses the same kinetics and the same inlet stream as the previous example. The reaction rate constants are slightly changed to: $k = [10\text{s}^{-1}, 1\text{s}^{-1}, 5\text{l/mol s}]$. The bounds on the concentrations, residence time of the reactors and the recycle ratio are the same. This changes the structure of the global solution, as well as the number and structures of the various local solutions as shown in Table 7. Again these are generated by solving the recycle PFR formulation to a local solution using multiple random starting points. Notice, that two of the local solutions actually contain PFRs with recycle streams. The global solution, however, does not contain a recycle.

This example was solved to global optimality (1 % relative convergence) using the tuned set of algorithmic parameters defined in Section 4.1. Both the full four component and the reduced component (including only A and C) formulations were used with and without the subnetwork split reformulations. The results of these runs are shown in Table 8. These results clearly illustrate the advantage of using the subnetwork forms. In the reduced component formulation, by introducing the subnetwork spilt, the solution time is reduced from over 7 CPU hrs. to about 45 CPU mins. Also evident is the advantage of including only the necessary components in the formulations. Removing components B and D from the formulation results in the solution time being cut in half, from 1.5 CPU hrs to 45 CPU mins.

Table 8. Global solution times for the Van De Vusse, Case 2 example.

Formulation	Sub-network	Iterations	CPU (min)
Recycle PFR	Full	26 524	681.37
	Subnetwork 1	3945	59.43
	Subnetwork 2	2305	50.45
Reduced Component	Full	35 229	437.10
	Subnetwork 1	4275	28.78
	Subnetwork 2	3035	19.24

4.3. SERIES-PARALLEL REACTION

This problem appears in Levenspiel (1973, p. 191) and is also studied by Achenie and Biegler (1990). The reaction scheme is defined by:

$$A + B \xrightarrow{k_1} R$$

$$R + B \xrightarrow{k_2} S$$

The rate expressions for this mechanism take the form:

$$f_1 = k_1\, c_A\, c_B$$
$$f_2 = k_2\, c_R\, c_B$$

where the reaction rate constant vector is defined as:

$$k = [68.80\text{l/molh},\ 34.40\text{l/mol h}].$$

The objective is to maximize the yield of the component R (C_R/C_A^0). The inlet to the network is a mixed feed of components A and B, both with a concentration of 0.5 mol/l. The bounds on the concentration were set to: $c_A, c_B, c_R, c_S \in [0, 0.5]$, and the bounds on the residence times were set to: $\tau^m, \tau^t \in [0.05\text{h}, 1.0\text{h}]$. Table 9 shows the local solutions for this example.

This example was solved to global optimality (1 % relative convergence) using the tuned set of algorithmic parameters. Both the full four component recycle PFR and the reduced component recycle PFR formulations were used with the subnetwork split. For this example, three components are needed in the formulation, A, B, and R. The results of these runs are shown in Table 10. Even though three out of the four components are needed in the reduced component formulation, there is a decrease in the required computational effort. The computational time was reduced nearly in half from 85 CPU min for both subnetworks in the four component form to 50 CPU min using three components.

Table 9. Local solutions for the Series-Parallel example.

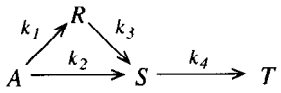
Obj. value	Configuration	Frequency (%)
0.500	PFR (0.4717 h)	79.8
0.458	CSTR (0.05 h) + PFR (0.6726 h)	2.7
0.343	CSTR (0.2396 h)	5.9
0	All Bypass	5.6
Failed	–	6.0 %

Table 10. Global solution times for the Series-Parallel Example

Formulation	Sub-network	Iterations	CPU (min)
Recycle PFR	Subnetwork 1	3915	81.84
	Subnetwork 2	187	3.12
Reduced component	Subnetwork 1	4342	48.36
	Subnetwork 2	137	1.33

4.4. LEVENSPIEL EXAMPLE

This example is taken from Levenspiel (1993 p. 8.21). The reaction scheme for this example takes the form:



The reaction expressions for this mechanism take the form:

$$f_1 = k_1\, c_A$$
$$f_2 = k_2\, c_A$$
$$f_3 = k_3\, c_R$$
$$f_4 = k_4\, c_S$$

where the reaction rate constant vector is defined as k = [0.21 s^{-1}, 0.20 s^{-1}, 4.2 s^{-1}, 0.004 s^{-1}]. The objective is to maximize the yield of component S, with an initial inlet to the network of pure A with a concentration of 1 mol/l. The bounds on the concentration were set to: $c_A, c_B, c_R, c_S \in [0, 1.0]$, and the bounds on the residence times were set to: $\tau^m, \tau^t \in [0.05\text{min}, 1.0\text{min}]$. Table 11 shows the local solutions for this example.

The problem was solved to global optimality using both the full four component formulation and a reduced component formulation containing only A, R, and S.

Table 11. Local solutions for the Levenspiel example.

Obj. value	Configuration	Frequency (%)
0.955	PFR (0.1922 min)	95.7
0.951	CSTR (0.05 min) + PFR (0.160 min)	2.6
0.824	CSTR (0.4226 min)	1.2
0	All bypass	0.1
Failed	–	0.4

Table 12. Global solution times for the Levenspiel example.

Formulation	Sub-network	Iterations	CPU (min)
Recycle PFR	Subnetwork 1	361	11.27
	Subnetwork 2	319	9.61
Reduced component	Subnetwork 1	5333	72.72
	Subnetwork 2	2498	31.88

A relative convergence tolerance of 0.1 % was used due to the small difference between local solutions. Table 12 gives the results for each of the two subnetwork forms. The results show something very interesting, the full four-component formulation takes an order of magnitude fewer iterations and 1 h less computational time to solve. This appears to be quite contradictory to all previous results. This discrepancy can be linked to the material balances around each of the reactors (Eqs. (19) – (22)). In the reduced component formulation, these balances cannot be written as equality constraints as they can be in the full component form. In this example, these balances have a dramatic effect on the solution. Table 13 shows results using the full component formulation when the material balances around the reactors are not included. Neither of the subnetwork forms was able to reach convergence in 10 000 iterations. For this example, due to the complex nature of the reaction mechanism, it is advantageous to include all four components in the formulation.

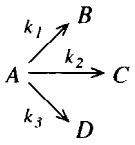
4.5. TRAMBOUZE REACTION

This reaction is originally presented by Trambouze and Piret (1959). It has also been studied by Achenie and Biegler (1986), Kokossis and Floudas (1990), and Schweiger and Floudas (1999). The scheme includes a zeroth-order, first-order,

Table 13. Global solution times for the Levenspiel example using the full component recycle PFR formulation, and removing the material balances around the reactors.

Subnetwork	Iterations	CPU (min)	Rel. difference (%)
1	10 000	191.05	4.67
2	10 000	170.21	0.82

and second-order reaction in parallel:



The reaction expressions for this mechanism take the form:

$$f_1 = k_1$$
$$f_2 = k_2\, c_A$$
$$f_3 = k_3\, c_A^2$$

where the reaction rate constant vector is defined as k = [0.025 mol/l min, 0.20 min^{-1}, 0.4 l/mol min]. The objective for this example is to maximize the selectivity of component C starting with a feed of pure A with a concentration of 1 mol/l. Converting the problem into the equivalent minimization, the objective function is written as:

$$\min \frac{-c_C^h}{1 - c_A^h} \tag{85}$$

The bounds on the concentration are set to: $c_A, c_B, c_R, c_S \in [0, 1.0]$, with the exception of c_A^h whose bounds are set to [0,0.9] to insure the objective function is defined over the full variable space. The bounds on the residence times were set to: $\tau^m, \tau^t \in [0.05\text{min}, 1.0\text{min}]$. Table 14 shows the local solutions for this example. It is interesting to note that one of the local solutions does include a recycle PFR, but the global solution does not. In this example, the system bypass stream, F^{ah} is removed from the network. Due to the nature of objective function, the inclusion of the bypass stream results in an infinite number of solutions with the same selectivity. By varying the split of the feed between the bypass stream and the CSTR and changing the size of the CSTR accordingly, the selectivity remains constant at 0.5.

The objective function, given by (85), is not linear nor is it convex. It is actually pseudo-convex which is the theoretical requirement for a convex optimization

Table 14. Local solutions for Trambouze example.

Obj. Value	Configuration	Frequency (%)
0.500	CSTR (0.125 h)	35.8
0.473	PFR (0.0587 h, R = 1.0)	41.0
Failed	–	23.2

problem. Therefore, the objective function need not be underestimated and can be included as is in both the upper and lower problem formulations. The drawback is since the objective function is not linear in nature, it cannot be included in the linear bounds updating problems. Another approach is to reformulate the objective function such that the resulting underestimated form is linear. This is accomplished by introducing the variable s, which represents the selectivity, into the formulation,

$$s = \frac{c_C^h}{1 - c_A^h}.$$

The objective function is then written as:

$$\min -s,$$

and the bilinear constraint,

$$s - s\, c_A^h - c_C^h = 0$$

is added to the formulation. As a result of this bilinear reformulation, the objective function is underestimated in a linear manner, and can be included in the linear bounds updating problems. The bounds on the additional variable s can be calculated from the bounds on $\boldsymbol{c}^h$:

$$s^L = \frac{c_C^{h,L}}{1 - c_A^{h,L}} \; s^U = \frac{c_C^{h,U}}{1 - c_A^{h,U}}$$

The example was solved to global optimality (1 % relative convergence) using the reduced component formulation with the subnetwork split. The only components needed in the formulation are A and C. Both forms of the objective function were explored. Table 15 presents results which show that the use of the pseudo-convex form of the objective function is preferred. The bilinear form can be included in the linear bounds updating problem, but the tightening of the variable bounds does not offset the relaxed nature of the objective function.

Table 15. Global solution times for the Trambouze example using the reduced component formulations with different forms of the objective function.

Obj. Form	Sub-network	Iterations	CPU (min)
Pseudo-convex	Subnetwork 1	421	2.02
	Subnetwork 2	2733	9.70
Bilinear	Subnetwork 1	425	1.93
	Subnetwork 2	4093	15.51

Table 16. Summary of the global solution times for all the examples presented.

Example	Local solutions	Iterations	CPU (min)
Van de Vusse, case 1	3	2405	12.01
Van de Vusse, case 2	5	7310	48.02
Series-Parallel	4	4479	49.69
Levenspiel	4	680	20.88
Trambouze	2	3154	11.72

4.6. SUMMARY OF COMPUTATIONAL RESULTS

Table 16 gives a summary of the computational results for the five example problems explored. The solution times provided are the total time and iterations required for both subnetwork forms using the tuned set of algorithmic conditions defined in Section 4.1. For all but the second Levenspiel example, the results shown were obtained using the reduced component formulations. In the second Levenspiel example, as shown in Section 4.4, it is advantageous to include all components in the formulation.

One of the more interesting results which deserves to be revisited is the effect of the problem formulation. In Example 1 the dramatic difference between each of the formulation classes was clearly illustrated. The effects of the various formulation improvements can be explained through the characteristics of the algorithm:

- *Additional redundant constraints* These allow for a tighter underestimating formulation. They restrict the allowable range of the bilinear terms which appear in various component balances. A tighter underestimating formulation directly results in a decrease in the number of iterations required to achieve convergence.
- *Recycle PFR formulation* By replacing the standard PFR and explicit recycle stream with a recycle PFR the number of nonconvex terms and variables are

decreased. This will also result in a reduction in the number of iterations required to reach convergence.

- *Subnetwork split* In reducing the problem to two subnetworks, the resulting formulations contain fewer constraints, variables, and nonconvex terms. Therefore, the combined time required to solve these two smaller problems is much less than that required to solve the full network.
- *Reduced component formulation* By removing a number of 'unimportant' components, the number of variables and bilinear terms is reduced. However, some of the material balances need to be rewritten as shown in Section 2.3.5. This results in a looser underestimating formulation since the number of degrees of freedom are increased (equality constraints were replaced with inequalities). In four of the five example problems, the reduction in the number of variables overcomes the loosing of the underestimator. However, in one example (the second Levenspiel example in section 4.4) it was better to include the full component space.

In general, each additional element aimed to reduce the number of nonconvex variables and terms in the formulation. These ideas can easily be applied to other problem classes solved using branch and bound type algorithms.

The following additional remarks about these examples can be made:

- All the local solutions to the examples differ in the structure of the network. There are no local solutions which are based on the same network structure with different reactor sizes or recycle ratios. This shows the nonconvexity in the mass network (the bilinearities in the mass balances) is the major complicating feature of the formulation.
- The proposed approach was actually able to identify the global solution within five iterations and less than 1 min of computational time for each of the examples. In most cases, the global solution was identified at the root node of the branch-and-bound tree requiring only a few seconds of computational time. The rest of the effort is required to prove the global optimality of the solution. Therefore, not only does the proposed approach offer a theoretical guarantee of convergence to the global minimum, but also acts as a highly effective directed search.

5. Conclusion

This paper presented a novel deterministic global optimization approach for the solution of the reactor network synthesis problem. A superstructure method was used to formulate the optimization as a nonconvex optimal control problem. The global solution to this formulation is determined using an algorithm developed around a branch-and-bound framework. A series of non-increasing upper bounds

on the global solution are obtained by solving the original formulation to a local optimum. A second series of non-decreasing lower bounds on the global solution are determined by solving a convex relaxation of the original problem. Convergence to the global solution is obtained by subdividing the region at each level of the branch-and-bound tree.

Numerous computational aspects of the approach were illustrated on a series of isothermal network problems taken from various literature sources. In every example, multiple local optima were observed, each possessing a different network structure. In terms of the global optimization approach, it was found that the problem formulation has the largest effect of the convergence rate of the algorithm. Using what can be referred to as a 'standard' formulation, the algorithm was unable to converge to the global solution. By performing various reformulations and adding redundant constraints, the algorithm was able to converge to the global solution with reasonable computational effort. Methods for the selection of branching variables and the updating of variable bounds were investigated. An adaptive scheme was developed to determine the branching variable based on the progress of the algorithm. A bounds updating method was presented which solves a series of simple linear optimization problems. Through the application of these methods, the solution time was reduced nearly in half. Convergence to the global optimum took on the order of minutes, however, the algorithm actually identified the global solution as the upper bound on the order of seconds.

Acknowledgements

The authors gratefully acknowledge financial support from the National Science Foundation.

References

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

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

Achenie, L.K.E. and Biegler, L.T. (1990), A Superstructure Based Approach to Chemical Reactor Network Synthesis. *Comput. Chem. Eng.* 14(1), 23.

Adjiman, C.S., Androulakis, I.P. and Floudas, C.A. (1998a), A Global Optimization Method, αBB, for General Twice–Differentiable NLPs – II. Implementation and Computational Results. *Comp. Chem. Eng.* 22(9), 1159–1178.

Adjiman, C.S., Androulakis, I.P., Floudas, C.A. and Neumaier, A. (1998b), A Global Optimization Method, αBB, for General Twice–Differentiable NLPs – I. Theoretical Advances. *Comp. Chem. Eng.* 22(9), 1137–1158.

Adjiman, C.S., Androulakis, I.P., Maranas, C.D. and Floudas, C.A. (1996), A Global Optimization Method, αBB, for Process Design. *Computers Chem. Eng., Suppl.* 20, S419–S424.

Adjiman, C.S. and Floudas, C.A. (1996), Rigorous Convex Underestimators for General Twice–Differentiable Problems. *Journal of Global Optimization* 9, 23–40.

Al-Khayyal, F.A. (1990), Jointly Constrained Bilinear Programs and Related Problems: An Overview. *Computers Math. Appl.* 19(11), 53–62.

Al-Khayyal, F.A. and Falk, J.E. (1983), Jointly Constrained Biconvex Programming. *Maths Ops Res.* 8, 273–286.

Androulakis, I., Maranas, C.D. and Floudas, C.A. (1995), αBB: A Global Optimization Method for General Constrained Nonconvex Problems. *Journal of Global Optimization* 7, 337–363.

Balakrishna, S. and Biegler, L.T. (1992a), A Constructive Targeting Approach for for the Synthesis of Isothermal Reactor Networks. *Ind. Eng. Chem. Res.* 31(9), 300.

Balakrishna, S. and Biegler, L.T. (1992b), Targeting Strategies for Synthesis and Energy Integration of Nonisothermal Reactor Networks. *Ind. Eng. Chem. Res.* 31(9), 2152.

Balakrishna, S. and Biegler, L.T. (1996), Chemical Reactor Network Targeting and Integration: An Optimization Approach. In: J.L. Anderson (ed.), *Advances in Chemical Engineering*, Vol. 23. Academic Press, pp. 247–300.

Chitra, S.P. and Govind, R. (1981), Yield Optimization for Complex Reactor Systems. *Chem. Eng. Sci.* 36, 1219–1225.

Chitra, S.P. and Govind, R. (1985), Synthesis of Optimal Serial Reactor Structures for Homogeneous Reactions. *AIChE J.* 31, 177–193.

Cordero, J., Davin, A., Floquet, P., Pibouleau, L. and Domenech, S. (1997), Synthesis of optimal reactor networks using mathematical programming and simulated annealing. *Comp. Chem. Engng.* 21, S46–S53.

CPLEX: 1997, Using the CPLEX Callable Library. ILOG, Inc.

Dyson, D.C. and Horn, F.J.M. (1967), Optimum Adiabatic Cascade Reactor with Direct Intercooling. *J. Opt. Theory Appl.* 1(1), 40–52.

Esposito, W.R. and Floudas, C.A. (2000a), Deterministic Global Optimization in Nonlinear Optimal Control Problems. *Journal of Global Optimization* 17, 97–126.

Esposito, W.R. and Floudas, C.A. (2000b), Global Optimization for the Parameter Estimation of Differential–Algebraic Systems. *I&EC Res.* 37(5), 1841–1858.

Feinberg, M. (1999), Recent results in optimal reactor synthesis via attainable region theory. *Chem. Eng. Science* 54, 2535–2543.

Feinberg, M. (2000a), Optimal reactor design from a geometric viewpoint – III. Critical CFSTRs. *Chem. Eng. Sci.* 55, 3553–3565.

Feinberg, M. (2000b), Optimal reactor design from a geometric viewpoint. Part II. Critical sidestream reactors. *Chem. Eng. Sci.* 55, 2455–2479.

Feinberg, M. and Hildebrandt, D. (1997), Optimal Reactor Design from a Geometric Viewpoint—I. Universal Properties of the Attainable Region. *Chem. Eng. Sci.* 52(10), 1637–1665.

Gill, P.E., Murray, W., Saunders, M. and Wright, M.H. (1986), User's Guide for NPSOL (Version 4.0). Systems Optimization Laboratory, Department of Operations Research, Stanford University. Technical Report SOL 86-2.

Glasser, D., Crowe, C. and Hildebrandt, D. (1987), The Attainable Region and Optimization in Concentration Spaces. *Ind. Eng. Chem. Res* 26(9), 1803–1810.

Hildebrandt, D. and Biegler, L.T. (1995), Synthesis of Reactor Networks. In: L.T. Biegler and M.F. Doherty (eds.), *Foundations of Computer Aided Process Design*, Vol. 91 of *AIChE Symposium Series*. New York, p. 52.

Hildebrandt, D. and Glasser, D. (1990), The Attainable Region and Optimal Reactor Structures. *Chem. Eng. Sci.* 45, 2161–2168.

Hildebrandt, D., Glasser, D. and Crowe, C.M. (1990), Geometry of the Attainable Region Generated by Reaction and Mixing: with and without Constraints. *Ind. End. Chem. Res.* 29, 49–58.

Horn, F. (1964), Attainable Regions in Chemical Reaction Technique. In: *The Third European Symposium on Chemical Reaction Engineering*. Pergamon, Oxford, pp. 1–10.

Horn, F.J.M. and Tsai, M.J. (1967), The Use of Adjoint Variables in the Development of Improvement Criteria for Chemical Reactors. *J. Opt. Theory Appl.* 1(2), 131–145.

Kokossis, A.C. and Floudas, C.A. (1990), Optimization of Complex Reactor Networks—I. Isothermal Operation. *Chem. Eng. Sci.* 45(3), 595–614.

Kokossis, A.C. and Floudas, C.A. (1991), Synthesis of Isothermal Reactor-Separator-Recycle Systems. *Chem. Eng. Sci.* 46(5/6), 1361–1383.

Kokossis, A.C. and Floudas, C.A. (1994a), Optimization of Complex Reactor Networks—II. Nonisothermal Operation. *Chem. Eng. Sci.* 49(7), 1037–1051.

Kokossis, A.C. and Floudas, C.A. (1994b), Stability in Optimal Design: Synthesis of Complex Reactor Networks. *AIChE J.* 40(5), 849–861.

Lakshmanan, A. and Biegler, L.T. (1996), Synthesis of Optimal Chemical Reactor Networks. *Ind. Eng. Chem. Res.* 35(4), 1344.

Levenspiel, O. (1973), *Chemical Reaction Engineering*. Wiley, New York.

Levenspiel, O. (1993), *The Chemical Reactor Omnibook*. OSU Book Stores, Inc.

Maranas, C.D. and Floudas, C. (1994), Global Minimum Potential Energy Conformations of Small Molecules. *Journal of Global Optimization* 4, 135.

Marcoulaki, E.C. and Kokossis, A.C. (1999), Scoping and Screening Complex Reaction Networks using Stochastic Optimization. *AIChE Journal* 45(9), 1977–1991.

McCormick, G.P. (1976), Computability of Global Solutions to Factorable Nonconvex Programs: Part I - Convex Underestimations Problems. *Mathematical Programming* 10, 147–175.

Ong, S.L. (1986), Optimization of CSTRs in Series by Dynamic Programming. *Biotechnology and Bioengineering* 28, 818–823.

Pontryagin, L.S. (1962), *Ordinary Differential Equations*. Addison-Wesley Publishing Co. (Translated from Russian by L. Kacinskas and W. B. Counts).

Quesada, I. and Grossmann, I. (1995), Global Optimization of Bilinear Process Networks with Multicomponent Flows. *Comp. Chem. Eng.* 19(12), 1219–1242.

Schweiger, C. and Floudas, C. (1999), Optimization Framework for the Synthesis of Chemical Reactor Networks. *I&EC Res.* 38(3), 744–766.

Schweiger, C.A. and Floudas C.A. (1998), MINOPT: A Modeling Language and Algorithmic Framework for Linear, Mixed-Integer, Nonlinear, Dynamic, and Mixed-Integer Nonlinear Optimization. Princeton University, 3rd edition.

Trambouze, P.J. and Piret E.L. (1959), Continuous stirred tank reactors: Designs for maximum conversions of raw material to desired product – Homogeneous reactions. *AIChE J.* 5, 384–390.